

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Um Processo Auto-Documentável de
Geração de Ontologias de Domínio
para Dados Semi-Estruturados**

por

SERGIO MEDEIROS SANTI

Dissertação submetida à avaliação,
como requisito parcial, para obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Carlos Alberto Heuser
Orientador

Porto Alegre, outubro de 2002.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Santi, Sergio Medeiros

Um Processo Auto-Documentável de Geração de Ontologias de Domínio para Dados Semi-Estruturados / por Sergio Medeiros Santi. – Porto Alegre: PPGC da UFRGS, 2002.

71 f.: il.

Dissertação (Mestrado) - Programa de Pós-Graduação em Ciência da Computação, Porto Alegre, BR-RS, 2002. Orientador: Heuser, Carlos Alberto.

1. Integração de esquemas. 2. Ontologias. 3. Dados Semi-Estruturados. 4. DTD. I. Heuser, Carlos Alberto. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^a. Wrana Maria Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fernsterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Sumário

Lista de Abreviaturas	4
Lista de Figuras	5
Lista de Tabelas	6
Resumo	7
Abstract	8
1 Introdução	9
2 Integração de Dados	11
2.1 Dados Estruturados	11
2.1.1 Ferramentas para Integração Automática de Esquemas	14
2.1.2 Trabalhos Futuros em Integração de Esquemas.....	15
2.2 Ontologias	16
2.2.1 Visão Geral Sobre Methontology	18
2.2.2 Aplicações de Ontologias a Dados Semi-estruturados	20
2.3 Dados Semi-Estruturados	21
2.3.1 Introdução ao XML	22
2.3.2 A Estrutura dos Dados XML	23
2.3.3 Esquemas para Documentos XML	23
2.3.4 A Interface de Programação de Aplicações (API).....	24
2.4 Síntese do Capítulo	24
3 A Proposta de Mapeamento e Integração	26
3.1 Visão Geral do Proposta	27
3.2 Convenções e Representação Gráfica	29
3.3 O Meta-modelo de Integração	31
3.3.1 Entidades.....	32
3.3.2 Relacionamentos.....	33
3.3.3 Thesaurus	36
3.4 Mapeamento e Integração	38
3.4.1 O Algoritmo de Mapeamento	38
3.4.2 O Algoritmo de Integração	39
3.4.3 Conflitos de Nomenclatura	40
3.4.4 Inferências e Conflitos de Cardinalidade.....	41
3.4.5 Estruturas Generalização/Especialização.....	42
3.5 Tratamento de Conflitos – Estudo de Caso	43
3.6 Intervenções do Usuário	48
3.7 Trabalhos Relacionados	50
3.8 Síntese do Capítulo	52
4 Exemplo de Referência	54
5 Conclusões e Trabalhos Futuros	61
Anexo 1 Apresentação do Protótipo	64
Bibliografia	68

Lista de Abreviaturas

API	Aplicação Program Interface
BD	Banco de dados
DD	Dicionário de dados
DOM	Document Object Model
DTD	Document Type Definition
E-R	Modelo Entidade Relacionamento
HOM	Has Ontology Merged
HOR	Has Ontology Remainder
HTML	Hiper Text Markup Language
IA	Inteligência Artificial
IGE	Integração Global de Esquemas
IRI	Interschema Relationship Identification
ISG	Integrated Schema Generation
MEQ	Mais Específico Que
MGQ	Mais Geral Que
ODE	Ontology Development Environment
PCData	Parsed Character Data
RDF	Resource Description Framework
SAX	Single API for XML
SBDF	Sistemas de Bancos de Dados Federados
SBDHD	Sistemas de bancos de Dados Heterogêneos e Distribuído
SGBD	Sistema Gerenciador de Banco de Dados
SGML	Standard Generalized Markup Language
SIDI	Distributed Intelligent Information Systems
SQL	Structured Query Language
UFRGS	Universidade Federal do Rio Grande do Sul
XML	eXtensible Markup Language
XSL	eXtensible Style Language
XSLT	eXtensible Style Language Transformations

Lista de Figuras

FIGURA 2.1 - Classificação das estruturas de dados quanto ao nível de estruturação	20
FIGURA 3.1 - Proposta de integração de fontes de informação semi-estruturada	28
FIGURA 3.2 - Mapeamento DTDs / pré-ontologias e integração pré-ontologias / ontologias	29
FIGURA 3.3 - Uma DTD para um memorando (a) e para uma carta (b)	29
FIGURA 3.4 - A representação gráfica da DTD de um memorando (a) e de uma carta (b) ...	30
FIGURA 3.5 - Pré-ontologias memorando (a), carta (b) e a ontologia integrada resultante(c)	31
FIGURA 3.6 - Meta-modelo para pré-ontologias e ontologias	32
FIGURA 3.7 - DTD (a), repres. gráfica (b) , pré-ontologia (c) e pré-ontologia atualizada (d)	33
FIGURA 3.8 - Instâncias para o meta-modelo referentes à pré-ontologia	36
FIGURA 3.9 - O modelo de dados do <i>thesaurus</i> e suas possíveis entradas	37
FIGURA 3.10 - Algoritmo de mapeamento da pré-ontologia a partir de uma DTD-XML	39
FIGURA 3.11 - Algoritmo de integração da pré-ontologia sobre a ontologia	40
FIGURA 3.12 - Cardinalidades incompatíveis	42
FIGURA 3.13 - Introdução de estruturas generalização/especialização	43
FIGURA 3.14 - Análise de relações para introdução de generalização/especialização	44
FIGURA 3.15 - Diferentes perspectivas (a e b) e o resultado da integração automática (c) ...	44
FIGURA 3.16 - Diferentes perspectivas (a e b) e o resultado da integração automática (c) ...	47
FIGURA 3.17 - Conflitos de nomenclatura(a e b), a integração automática(c) e outras(d e e)	47
FIGURA 3.18 - Conflitos de nomenclatura(a e b) e o resultado da integração automática (c)	48
FIGURA 4.1 - Representação textual das DTDs <i>Conference</i> (a) e <i>Workshop</i> (b)	55
FIGURA 4.2 - Representação gráfica das DTDs <i>Conference</i> (a) e <i>Workshop</i> (b)	55
FIGURA 4.3 - Pré-ontologias <i>Conference</i> (a) e <i>Workshop</i> (b) mapeadas automaticamente ...	56
FIGURA 4.4 - Pré-ontologias <i>Conference</i> (a) e <i>Workshop</i> (b) atualizadas pelo usuário	56
FIGURA 4.5 - Ontologia resultante da integração da pré-ontologia <i>Conference</i>	57
FIGURA 4.6 - Ontologia após sua integração com a pré-ontologia <i>Workshop</i>	68
FIGURA 4.7 - Ontologia após a introdução da estrutura generalização/especialização	69
FIGURA 4.8 - Visão pública da ontologia resultante do processo de integração	70
FIGURA 4.9 - Decomposição de consulta a ontologia em termos das fontes	71
FIGURA A.1 - Utilização do método <code>getParseTree()</code> sobre uma DTD-XML	65
FIGURA A.2 - Sub-ontologias cadastro e histórico e a ontologia local resultante	66
FIGURA A.3 - Propriedades do conceito aluno	67
FIGURA A.4 - <i>Thesaurus</i>	68

Lista de Tabelas

TABELA 2.1 - Exemplos de conceitos utilizados em ontologias	15
TABELA 2.2 - Relação binária emprega (a) e a relação binária afiliado (b)	17
TABELA 2.3 - Atributos da instância fotografia (a) e da instância peso (b)	17
TABELA 2.4 - Axioma estudante de PhD (a) e líder de projeto (b)	18
TABELA 2.5 - Constantes do domínio publicações	18
TABELA 2.6 - Fórmula para normalização de publicações	18
TABELA 2.7 - Instâncias do domínio Pessoas	19
TABELA 3.1 - Componentes visuais utilizados para representar pré-ontologias e ontologias	31
TABELA 3.2 - Exemplos de entradas do <i>thesaurus</i>	38
TABELA 3.3 - A tabela <i>Concept</i> instanciada	45
TABELA 3.4 - A tabela <i>Relationship</i> instanciada	46
TABELA 3.5 - A tabela <i>Integration</i> instanciada	46
TABELA 3.6 - Instantâneo parcial do <i>thesaurus</i>	47
TABELA 4.1 - Entradas do <i>thesaurus</i> necessárias ao mapeamento e integração	56
TABELA 4.2 - Instantâneo da tabela <i>integration</i>	57

Resumo

Dados são disponibilizados através dos mais distintos meios e com os mais variados níveis de estruturação. Em um nível baixo de estruturação tem-se arquivos binários e no outro extremo tem-se bancos de dados com uma estrutura extremamente rígida. Entre estes dois extremos estão os dados semi-estruturados que possuem variados graus de estruturação com os quais não estão rigidamente comprometidos. Na categoria dos dados semi-estruturados tem-se exemplos como o HTML, o XML e o SGML. O uso de informações contidas nas mais diversas fontes de dados que por sua vez possuem os mais diversos níveis de estruturação só será efetivo se esta informação puder ser manejada de uma forma integrada e através de algum tipo de esquema.

O objetivo desta dissertação é fornecer um processo para construção de uma ontologia de domínio que haja como esquema representativo de diferentes conjuntos de informação. Estes conjuntos de informações podem variar de dados semi-estruturados a dados estruturados e devem referir-se a um mesmo domínio do conhecimento. Esta proposta permite que qualquer modelo que possa ser transformado no modelo comum de integração possa ser utilizado com entrada para o processo de integração.

A ontologia de domínio resultante do processo de integração é um modelo semântico que representa o consenso obtido através da integração de diversas fontes de forma ascendente (*bottom-up*), binária, incremental, semi-automática e auto-documentável. Diz-se que o processo é ascendente porque integra o modelo que representa a fonte de interesse sobre a ontologia, é binário porque trabalha com dois esquemas a cada integração o que facilita o processo de documentação das integrações realizadas, é incremental porque cada novo esquema de interesse é integrado sobre a ontologia vigente naquele momento, é semi-automático porque considera a intervenção do usuário durante o processo e finalmente é auto-documentável porque durante o processo, toda integração de pares de conceitos semanticamente equivalentes é registrada. O fato de auto-documentar-se é a principal característica do processo proposto e seu principal diferencial com relação a outras propostas de integração.

O processo de mapeamento utiliza, dos esquemas de entrada, toda a informação presente ou que possa ser inferida. Informações como se o conceito é léxico ou não, se é raiz e os símbolos que permitem deduzir cardinalidades são consideradas. No processo de integração são consideradas práticas consagradas de integração de esquemas de BDs, na identificação de relacionamentos entre objetos dos esquemas, para geração do esquema integrado e para resolução de conflitos.

As principais contribuições desta dissertação são (i) a proposta de um meta-modelo capaz de manter o resultado dos mapeamentos e das integrações realizadas e (ii) a especificação de um processo auto-documentável que de sustentação a auditoria do processo de integração.

Palavras-chave: Integração de esquemas, Ontologias, Dados Semi-Estruturados.

TITLE: “A SELF-DOCUMENTED PROCESS OF DOMAIN ONTOLOGIES GENERATION FOR SEMI-STRUCTURED DATA”

Abstract

Data are available through the most different means and with the most varied structuring levels. In a low level of structuring it is had binary files and in the other end it is had databases with its extremely rigid structure. Between these two ends they are the semi-structured data that possess varied structuring degrees with which are not strictly committed. In the category of the semi-structured data it is had examples as HTML, XML and SGML. The use of information contained in the most several sources of data that possess the most several structuring levels for its time will only be effective if this information can be handled in an integrated way and through some type of schema.

The objective of this dissertation is to supply a process for construction of domain ontology that act as representative schema of different groups of information. These groups of information can vary of semi-structured data to structured data and they should refer to a same domain of the knowledge. This proposal allows that any model that can be transformed in the common model of integration can be used with entrance for the integration process.

The domain ontology resulting of the integration process is a semantic model that represents the consent obtained through the integration of several sources in an ascending way (bottom-up), binary, incremental, semiautomatic and self-documented. It is said that the process is ascending because it integrates the model that represents the source of interest on the ontology, it is binary because it works with two schemata to each integration that facilitates the process of documentation of the accomplished integrations, it is incremental because each new schema of interest is integrated on the effective ontology in that moment, it is semiautomatic because it considers the user's intervention during the process and finally it is self-documented because during the process, all integration of pairs of concepts semantically equivalent is registered. The fact of to be self-documented is the main characteristic of the proposed process and its main differential with relationship the other integration proposals.

The mapping process uses all the present information or information that can be inferred from the entry schemata. Information as, if the concept is lexical or not, if the concept is root or not and the schema's symbols that allow to deduce cardinalidades are considered. In the integration process are considered consecrated practices of databases schemata integration, in the identification of relationships among objects of the schemata, for generation of the integrated schema and for conflicts resolution.

The main contributions of this dissertation are (i) the proposal of a goal-model capable to maintain the result of the mappings and of the accomplished integrations and (ii) the specification of a self-documented process that allows to audit the integration process.

Keywords: Schema Integration, Ontologies, Semi-Structured Data.

1 Introdução

Dados semi-estruturados, bem como, a necessidade de integrá-los tem ocupado um grande espaço na comunidade científica desde o início das pesquisas com hipertexto, HTML, XML mais recentemente, e das aplicações destas tecnologias na *Web* [ABI 97]. Desde que primeiro os meios acadêmicos e depois as grandes massas de usuários passaram a ter acesso a *Web* houve a necessidade de adotar alguma forma de resumir as centenas, ou milhares de informações existentes como forma de reduzir o volume de informação a ser analisado, bem como incrementar a qualidade do resultado final.

A principal motivação deste trabalho derivou-se da constatação, com base na literatura consultada e relacionada neste trabalho, de que tanto no que se refere a integração de dados estruturados (esquemas de BDs convencionais) quanto na de integração de dados semi-estruturados não há trabalhos com a preocupação de manter o registro do histórico das operações realizadas para gerar o esquema integrado que no caso deste trabalho é uma ontologia de domínio. Assim além de fornecer uma ontologia a usuários e agentes de software este trabalho propõe um processo que mantém este histórico.

O fato de registrar o histórico de integração é a principal característica deste trabalho e o que o distingue dos demais. Através desta característica é possível, por exemplo: (i) definir a estratégia de evolução da ontologia motivada pelas alterações nos esquemas componentes ou pela inclusão de novos esquemas a integrar; (ii) realizar a análise qualitativa da ontologia gerada através do estudo do histórico de integração e; (iii) utilizar as decisões do usuário que foram registradas durante o processo como forma de automatizar novos processos de integração.

Este trabalho propõe a geração e utilização de uma ontologia como um meta-esquema que represente um determinado domínio do conhecimento [STU 99]. Este meta-esquema pode ser usado por agentes humanos e de software [GRU 93] que tanto podem atuar na construção do meta-esquema, quanto o utilizar, extraíndo dele a informação necessária para seus propósitos específicos. Esta ontologia será obtida de documentos XML [SIL 02, COV 02 e W3C 02], parte relevante do universo semi-estruturado e que tem concentrado muito dos atuais esforços de pesquisa e desenvolvimento quanto a sua padronização e quanto a criação de novas aplicações.

Entre as possíveis aplicações deste trabalho estão seu uso: (i) junto a agentes humanos e de software que precisem **compreender** um dado domínio semi-estruturado para inferir relações não expressas literalmente; (ii) em processos de extração de dados que precisem **conhecer** os conceitos envolvidos e as relações entre eles; (iii) como forma de construir modelos para dados XML permitindo que seus usuários compartilhem o conhecimento do domínio por eles representados.

O contexto do processo proposto, constitui-se de um conjunto de DTDs que se referem a um dado domínio do conhecimento. Deste conjunto de DTDs deseja-se obter uma pré-ontologia que represente todos os conceitos destas DTDs em um modelo comum de representação o que será obtido pela aplicação de um algoritmo a cada DTD. A ontologia será obtida, a partir de um processo ascendente (*bottom-up*), incremental e binário no qual cada pré-ontologia é integrada uma a uma sobre a ontologia. Desta forma a ontologia passa a representar um consenso obtido pela integração dos conceitos presentes em cada DTD participante do processo. Este consenso é uma das características fundamentais das ontologias e o que a faz estar presentes neste e em diversos outros trabalhos [BER 99, DEC 98, DOR 00,

DUS 97, BEM 99, FAN 91 FEN 99, GEN 97, MEL 02, STU 00] onde esta característica se faz necessária.

No processo de integração, como aliás ocorre no processo de integração de esquemas de bancos de dados convencionais, previsivelmente ocorrerão conflitos semânticos, como palavras sinônimas representando diferentes conceitos, palavras homônimas representando o mesmo conceito, antônimas significando palavras ou locuções de significação oposta e finalmente homógrafas que representam vocábulos que têm a mesma grafia, mas significações diferentes [BAT 86, SHE 90 e BAT 92]. Assim, é necessário que estes conflitos sejam, sempre que possível, resolvidos automaticamente o que será feito através de práticas já consagradas no processo de integração de esquemas de bancos de dados [FAN 91 e BAT 86]. Aqueles conflitos que não puderem ser resolvidos automaticamente deverão ser resolvidos por um especialista no domínio de problema em questão [BOU 99], caracterizando assim, a natureza semi-automática do processo de integração proposto. Deve-se considerar, contudo, que mesmo que não houvesse conflitos a resolver a intervenção do especialista seria uma forma de homologar o modelo gerado, agregando-lhe maior confiabilidade.

Conforme citado até aqui diversos trabalhos na área de integração de informação influenciaram esta proposta. O projeto SIDI (Modelling and Development of Distributed Intelligent Information Systems [CAS 99] tinha o objetivo de integrar informações provenientes de diversas instituições de saúde preservando a autonomia das fontes. O projeto considerava *thesaurus*, esquemas de exportação, modelo canônico, resolução de conflitos e mapeamento entre objetos dos esquemas fonte e objetos do esquema comum, entre outros. Este projeto utiliza DTDs-XML como esquema de exportação também garantindo a autonomia das fontes; utiliza um modelo E-R como modelo canônico; considera metodologias de resolução de conflitos, utiliza um processo de integração que gera o mapeamento dentre conceitos da ontologia (esquema integrado) e os conceitos presentes nas fontes e faz uso extensivo de *thesaurus* com entradas provenientes de fontes distintas.

No âmbito do grupo de pesquisa em banco de dados da UFRGS, diversos trabalhos de mestrado e doutorado concluídos ou em curso consideram o uso de ontologias como exposto acima em maior ou menor grau: Extração de dados semi-estruturados [DOR 00]; Materialização de dados semi-estruturados; Visualização de dados semi-estruturados; Uma Abordagem *Bottom-Up* para a Integração Semântica de Esquemas XML [MEL 02]. Este último, uma tese de doutorado recém concluída que se desenvolveu juntamente com esta dissertação, propõe a integração semântica de esquemas XML, através de uma abordagem bottom-up.

Este trabalho está organizado conforme segue. No capítulo 2 é feita uma abordagem teórica sobre trabalhos existentes sobre os quais esta proposta se fundamenta, ou seja, integração de esquemas de bancos de dados convencionais, ontologias e dados semi-estruturados. Já no capítulo 3 a proposta de mapeamento e integração, seu modelo de dados e algoritmos entre outros são apresentados e discutidos. O capítulo 4 apresenta um exemplo de referência, onde são mostrados os esquemas (DTDs) a mapear, o resultado de seu mapeamento para as pré-ontologias e a sua integração que dá origem a ontologia. No capítulo 5, são apresentadas as conclusões e trabalhos futuros a serem desenvolvidos. Finalmente o anexo 1 apresenta uma aplicação experimental, um protótipo elaborado como parte de um trabalho de graduação [MAR 00], construída para exercitar e validar a proposta, como ela se apresentava naquele momento.

2 Integração de Dados

Neste capítulo serão discutidos pontos relevantes a este trabalho encontrados na literatura. Primeiramente são abordadas técnicas de integração de esquemas representativos de dados estruturados e que podem ser utilizadas na integração de semi-estruturados. A seção seguinte aborda ontologias, onde é apresentada uma metodologia que influenciou a criação do meta-modelo apresentado no capítulo 3 e outras implementações que integram informação semi-estruturada através de ontologias. Finalmente são discutidos aspectos dos documentos XML importantes à compreensão do processo de mapeamento proposto por este trabalho.

2.1 Dados Estruturados

O objetivo desta seção é apresentar aspectos das técnicas de integração de esquemas para dados estruturados que podem ser usados para integrar esquemas para dados semi-estruturados. Para isto, esta seção aborda dois importantes enfoques de integração, ou seja a integração geral de esquemas e os bancos de dados federados. As causas da heterogeneidade representacional e uma taxonomia de conflitos são discutidas, uma vez que é necessário classificar os conflitos para então resolvê-los. Esta informação é então utilizada por técnicas de identificação de relacionamentos entre esquemas e por técnicas de geração do esquema integrado. Finalmente, são discutidas algumas ferramentas automatizadas disponíveis na literatura e pontos que ainda requerem pesquisa e que focam, principalmente, as técnicas de identificação de relacionamentos entre esquemas.

O assunto integração de dados não é um tópico de pesquisa recente e tem sido objeto de interesse quando diversas fontes de informação fornecem dados em diferentes formatos. Assim o assunto tem estado em voga há muito tempo por organizações que possuem bancos de dados distribuídos de forma lógica e física. Mais recentemente com a disponibilidade de grande quantidade de informação semi-estruturada na *Web* esta situação se agravou. O objetivo desta seção é estudar soluções propostas para integração de esquemas de bancos de dados heterogêneos, distribuídos, autônomos e interoperáveis como fomento a uma proposta de integração de esquemas extraídos de fontes de dados semi-estruturados.

Sistemas de informação que se relacionem e operem com diferentes graus de heterogenia, distribuição, autonomia e interoperabilidade são segundo Elmagarmid, Du e Ahmed [ELM 99], conhecidos pela designação genérica de sistemas de bancos de dados heterogêneos e distribuídos (SBDHDs) e são classificados em três categorias: Integração global de esquemas; Bancos de dados federados e linguagens para bancos de dados múltiplos. As duas primeiras abordagens serão sucintamente discutidas a seguir.

A **integração global de esquemas (IGE)** foi uma das primeiras tentativas de compartilhar dados entre SBDHDs e é baseada na integração completa dos múltiplos bancos de dados envolvidos, de forma a proporcionar uma visão unificada através de um esquema também unificado (esquema global). Batini, Lenzerini e Navathe [BAT 86] apresentam uma comparação entre diversas metodologias para integração de esquemas. A principal vantagem desta abordagem é que para os usuários do esquema global é fornecida uma visão dos dados consistente e uniforme e que os libera dos aspectos referentes a distribuição e heterogeneidade, fazendo com que múltiplos bancos de dados apareçam como um único banco de dados. Apesar das claras vantagens, a IGE apresenta alguns inconvenientes.

- Como não há uma solução geral para os problemas de integração advindos de conflitos semânticos, estruturais e comportamentais o processo é difícil e dependente do usuário.
- O processo de resolução de conflitos reduz a autonomia entre as fontes.
- Havendo mais que dois esquemas a integrar, pode-se adotar duas abordagens: ou utilizam-se todos os esquemas no processo de integração ou integram-se os esquemas dois a dois. A primeira tem a vantagem de considerar todo o conhecimento semântico e a desvantagem de tornar o processo complexo. Já a segunda alternativa facilita o processo pela redução do volume, mas trás a desvantagem de analisar informações parciais a cada par de esquemas analisados. A segunda abordagem também trás contra si o fato de que a ordem em que os esquemas são integrados afeta o resultado final, o que não é desejável.

O grande diferencial da **arquitetura de bancos de dados federados** (SBDF) em comparação com a IGE, é que não há a necessidade de um esquema global estático de integração. A integração não precisa ser completa desde que atenda as necessidades de seus usuários. Tipicamente um SBDF utiliza um modelo de dados comum, uma linguagem interna de comando, alguns esquemas e processadores [BOU 99]¹.

Num SBDF, a administração centralizada dá condições de que seja criado um esquema federado simples que se assemelharia muito da IGE. Esta estratégia contudo implica em pelo menos um inconveniente, ou seja, a administração centralizada se contrapõe a autonomia local que fica reduzida.

Quando se trata de integração de dados estruturados, a discussão do problema da diversidade representacional torna-se importante. Segundo Batini, Lenzerini e Navathe [BAT 86] existem basicamente três **causas para a heterogeneidade representacional**:

- Diferentes perspectivas e necessidades: este problema ocorre quando, na modelagem, diferentes grupos de usuários adotam diferentes pontos de vista para a mesma informação.
- Construtores equivalentes: os construtores utilizados na criação de modelos permitem diversas possibilidades de modelagem. Assim diferentes construções podem ser utilizadas para modelar o mesmo domínio de forma equivalente.
- Especificações de projeto incompatíveis: diferentes especificações de projeto resultam em diferentes esquemas. A especificação de cardinalidades é especialmente sensível a este problema.

Sheth e Larson [SHE 90], apresentam uma **taxonomia de conflitos de representação** que foi elaborada com base em Batini, Lenzerini e Navathe [BAT 86]. Esta classificação compreende diversos fatores que não afetam diretamente este trabalho, mas que são descritos em detalhe pelos autores citados. Dos conflitos de representação de interesse a esta proposta estão os conflitos semânticos e os estruturais.

Os **conflitos semânticos** se dividem em: (i) homonímia que ocorre quando o mesmo rótulo ou rótulos semelhantes em esquemas distintos representam objetos distintos; (ii) sinonímia que ocorre quando rótulos distintos representam o mesmo conceito em locais distintos; (iii) restrições de modelagem ocorrem quando objetos equivalentes são agrupados diferentemente em locais distintos e (iiii) conflitos descritivos ocorrem quando o mesmo objeto é modelado de forma distinta em diferentes locais.

Os **conflitos estruturais**, ainda em Sheth e Larson [SHE 90], subdividem-se em: (i) conflitos de estrutura que ocorrem quando em diferentes esquemas locais o mesmo

¹ A referência citada contém uma descrição bastante detalhada destes componentes.

objeto é modelado através de diferentes estruturas; (ii) conflitos de agregação que ocorrem quando o mesmo objeto é agrupado de forma diferente em diferentes esquemas locais; (iii) conflito de domínio de atributo que ocorre quando um atributo de um objeto é valorado diferentemente em diferentes esquemas locais, por exemplo, diferenças de unidade, de precisão, de codificação, de granularidade e; (iiii) conflitos comportamentais que ocorrem quando diferentes locais impõe diferentes restrições de integridade, métodos e transições de estado para um mesmo objeto, fazendo com que apresentem comportamentos diferentes em diferentes locais.

Batini, Ceri e Navathe [BAT 92] introduzem a figura do dicionário de dados (DD) como ferramenta de integração. O processo de construção do DD se dá em três etapas: a) selecionar os esquemas a integrar; b) integrar os esquemas locais resolvendo os problemas de integração semântica e; c) refinar o esquema integrado.

Batini, Lenzerini e Navathe [BAT 86], dizem que a integração dos esquemas locais é feita através de um esquema global gerado por: a) comparar; b) adequar; c) integrar e; d) reestruturar esquemas. A parte crítica do processo é identificar conflitos e definir um método para resolvê-los. Os relacionamentos entre representações podem ser: 1) idênticos; 2) similares; 3) compatíveis e; 4) incompatíveis, sendo que haverá conflito sempre que não forem idênticos.

Dentre a literatura consultada, um dos trabalhos mais expressivos, no que tange a como **resolver a heterogeneidade semântica**, está no trabalho de D. Fang et al [FAN 91] que propõe uma arquitetura chamada Remote-Exchange que suporta o compartilhamento controlado de informações entre SBDHDs. Esta arquitetura se compõe do: a) núcleo de objetos do modelo de dados b) da linguagem de compartilhamento remoto, c) do léxico local e; d) do dicionário semântico.

A parte da arquitetura de Fang et al [FAN 91] que mais chama a atenção do ponto de vista deste trabalho é o léxico local. Ele se constitui de uma coleção estática de termos que representam conhecimento no formato <termo>descriptor_de_relacionamento <termo>, onde o termo da esquerda é um conceito desconhecido expresso na forma de seu relacionamento com o termo conhecido da direita. Entre os descritores de relacionamentos, segundo a proposta de Fang, estão: identical, equal, compatible, kindof, assoc, collection of, instance of, common, feature e has.

Uma lista de conceitos consensuais é chamada de ontologia² e é distinta para diferentes domínios de aplicação. Os diversos pacotes de ontologias são armazenados em dicionários semânticos. Na arquitetura de Fang et al [FAN 91], um dicionário semântico é um repositório global que contém relacionamentos entre termos presentes em diferentes léxicos locais. Os diversos componentes identificados são agrupados em uma hierarquia de conceitos.

Segundo Ram e Ramash [RAM 99] as abordagens importantes com relação a integração de bancos de dados heterogêneos são a de esquema global, onde os esquemas locais são combinados em um esquema integrado e de esquemas federados, onde cada BD local provê um esquema de exportação. E a **integração de esquemas** é o núcleo de metodologias que usam estas abordagens para proporcionar interoperabilidade entre BDs heterogêneos.

² Mais tarde neste capítulo, na seção 2.2 que trata do assunto, o termo ontologia será definido.

Uma metodologia típica para integração de esquemas [RAM 99 e BON 94] pode ser dividida em quatro fases. 1) Traduzir os esquemas para um modelo de dados comum, como o modelo E-R por exemplo. 2) Identificar relacionamentos entre os objetos dos diversos esquemas a integrar. 3) Gerar o esquema integrado e conseqüentemente resolver os conflitos semânticos existentes. 4) Gerar o esquema de mapeamento, o que significa manter informação sobre os mapeamentos do esquema integrado com objetos dos esquemas locais.

As técnicas de **identificação de relacionamentos entre esquemas** (interschema relationship identification – IRI) baseadas em esquemas conceituais usam um processo em duas etapas que se constituem de (i) identificar os objetos que são relacionados e (ii) classificar os relacionamentos entre estes objetos. Ramash e Ram [RAM 95] sugerem que as propriedades de todos os objetos sejam usadas para determinar relacionamentos entre objetos do BD. Por exemplo, entidades podem ser comparadas através de seus nomes e pela descrição de seu papel, relacionamentos podem ser comparados através de seus nomes, cardinalidades e a similaridade com outros objetos. Para possibilitar estas comparações é necessário utilizar dicionários e *thesaurus*.

O objetivo de analisar esquemas é identificar objetos que são semanticamente relacionados, entretanto, é necessário não só identificar mas também classificar os relacionamentos entre estes objetos. Esta classificação depende da metodologia usada. As mais importantes, dentro do contexto deste trabalho, são citadas a seguir.

Larson, Navathe, Elmasri [LAR 89] sugerem quatro tipos de equivalência entre dois atributos A e B, ou sejam, *A equal B*, *A contains B*, *A contained-in B*, e *A overlap B*. Eles também definem cinco tipos de relacionamentos entre entidades e relacionamentos, os quais podem ser, *A equal B*, *A contains B*, *A contained-in B*, *A overlap B* e *A disjoint B*. Os usuários devem especificar estes relacionamentos para cada entidade e para cada relacionamento existente no modelo.

DeSouza [DES 86], Hayne e Ram [HAY 90] e Ramash e Ram [RAM 95] descrevem relacionamentos entre objetos em termos de graus de similaridade e dissimilaridade. Esta classificação é interessante para automação do processo de determinação de relacionamentos entre esquemas. Assume-se que os objetos e seus graus de similaridade serão oferecidos ao integrador de esquemas para a geração de relacionamentos.

As duas seções a seguir abordam respectivamente as iniciativas que culminaram em algum tipo de ferramenta que automatizam, em variados graus, a integração de esquemas e as lacunas deixadas pelos trabalhos de integração até então realizados e que apontam para aspectos que podem ser aprimorados.

2.1.1 Ferramentas para Integração Automática de Esquemas

Um dos primeiros esforços para automatizar fases do processo de integração foi o trabalho de DeSouza [DES 86], que tem seu foco na identificação de relacionamentos entre objetos dos esquemas a integrar. O autor apresenta um sistema especialista projetado para integrar esquemas conceituais que utiliza um conjunto de funções para comparar objetos nos esquemas envolvidos. Estas funções usam o nome e a estrutura para estimar as semelhanças entre objetos. Cada função tem um peso que indica a importância relativa desta semelhança para o usuário. Valores de similaridade acima de um dado valor são apresentados ao usuário como provavelmente similares.

Sheth et al [SHE 88] apresenta uma ferramenta que conduz o usuário por um processo de integração em cinco etapas: a) coleta de informações sobre o esquema; b) criação e deleção de classes equivalentes (entidades e categorias); c) criação e deleção de classes equivalentes (relacionamentos); d) assertivas de usuário (entidades e categorias) e; e) assertivas de usuário (relacionamentos). Os usuários fornecem informações sobre objetos que eles acreditam poderem ser integrados e recebem de volta uma lista com pares de objetos em ordem de probabilidade quanto a integração, o usuário então, atribui a estes relacionamentos um dos seguintes: *equal*, *contined-in*, *contains*, *disjoint-but-integratable* e *disjoint-and-nonintegratable*. A grande deficiência desta técnica é o grande envolvimento com o usuário e a pouca flexibilidade para capturar informação semântica.

A ferramenta apresentada por Ramash e Ram [RAM 95] utiliza conhecimento sobre entidades, atributos e relacionamentos para gerar valores de similaridade entre entidades, atributos e relacionamentos. Heurísticas são usadas para reduzir o universo de pesquisa. Nesta ferramenta a interação com o usuário é vista como uma fonte adicional de conhecimento, as outras duas fontes são um mecanismo de identificação de relacionamentos e um mecanismo de geração do esquema integrado.

2.1.2 Trabalhos Futuros em Integração de Esquemas

A despeito do fato de pesquisadores estarem estudando o tema integração de esquemas desde o início dos anos 80, a natureza complexa do problema tem deixado diversos pontos sem solução, assim, nesta seção, serão apresentadas algumas direções para pesquisa futura. As idéias a seguir foram classificadas como sendo aplicáveis ao processo de identificação de relacionamentos entre esquemas (IRI) ou de geração do esquema integrado (ISG).

No intuito de melhorar o nível de automação da identificação de relacionamentos entre esquemas é preciso evoluir das técnicas atuais que utilizam apenas a informação presente nos esquemas para abordagens que combinem informação de múltiplas fontes para inferir relacionamentos que melhor reflitam sua semântica.

Ramash e Ram [RAM 97] apresentam uma abordagem que descreve como a semântica comunicada pelas restrições de integridade pode ser usada em IRI. Eles introduzem o conceito de relacionamento baseado em restrições entre objetos da base de dados e apresentam uma metodologia para geração destes relacionamentos. As restrições de integridade dos esquemas a serem integrados são analisadas para gerar os relacionamentos entre esquemas dos objetos envolvidos nestas restrições.

Outra abordagem recente de IRI é utilizar a semântica comunicada pelos valores de dados. Técnicas de *data mining* podem ser estendidas para uso no contexto de IRI. Existem trabalhos que descobriram restrições de integridade semântica a partir da análise dos valores de dados existentes no BD. Outros pesquisadores utilizam a teoria de recuperação lingüística e de informações para auxiliar a identificação de objetos, onde taxonomias dos conceitos presentes nos esquemas são construídas e em conjunto com dicionários e *thesaurus* são utilizadas para resolver ambigüidades.

As técnicas de geração de esquemas integrados (ISG) com foco na resolução de diferenças estruturais encontram-se em um estágio maduro. Contudo, um dos problemas mais desafiadores a serem resolvidos nesta área é o gerenciamento da evolução de esquemas que precisam mudar em resposta as mudanças nos esquemas locais. As técnicas atuais assumem

que esquemas locais são basicamente estáticos e que mudanças são infreqüentes, o que nem sempre é verdade.

2.2 Ontologias

Esta seção descreve o que são ontologias no contexto deste trabalho. Para isso será feita uma breve introdução procurando situar cronologicamente o assunto. Dentro da bibliografia estudada, Methontology mereceu uma atenção especial em virtude de descrever claramente e em termos de tabelas relacionais as classes de objetos, atributos, relações, funções, entre outros pontos de interesse e que foram extremamente úteis quando da elaboração do meta-modelo utilizado nesta abordagem e descrito detalhadamente no capítulo 3. Em função disto uma visão geral sobre esta metodologia será apresentada. Finalmente será apresentado, também na forma de uma visão geral, cinco aplicações de ontologias, os projetos Observer, OntoBroker, On2Broker, Momis e Infomaster.

“O grande interesse sobre o assunto deve-se, em grande parte ao que as ontologias se propõem, ou seja: Fornecer uma compreensão comum e compartilhada de algum domínio que pode ser comunicada entre pessoas e computadores.” [STU 99]

No contexto das ciências da computação a área de inteligência artificial (IA) vem utilizando ontologias para que módulos de software simulem compreender significados através da apresentação de resultados que traduzam esta compreensão. Em IA bases de conhecimento ontológico são conjuntos de especificações usados para criar compromissos ontológicos e tem sido construídos para especificar conceitos que possam ser compartilhados e reutilizados por módulos de software no intuito de conferir-lhes uma classe comum de comportamentos e ainda inferir resultados não expressos literalmente [GRU 93]. Ainda na área de IA Farquhar, Fikes e Rice [FAR 96] dizem: “Para que um agente faça declarações e consultas sobre um domínio, ele precisa usar uma conceituação daquele domínio. Uma conceituação de domínio nomeia e descreve as entidades que podem existir naquele domínio, bem como, as relações entre eles”.

Especificações explícitas para conceituar um domínio, chamadas ontologias, são essenciais para o desenvolvimento e uso de sistemas inteligentes e para a interoperabilidade de sistemas heterogêneos. Ontologias informam ao usuário de um sistema inteligente o vocabulário disponível para interação, a que domínio se refere e que significado tem seus termos. Ontologias também habilitam a interoperabilidade entre agentes, uma vez que suas interações mais significativas se dão quando eles compartilham uma interpretação comum para o vocabulário utilizado em sua comunicação. Por fim, ontologias podem servir para incorporar e referenciar o consenso alcançado por uma comunidade profissional (por exemplo, na medicina), quanto ao significado de seu vocabulário técnico que é usado em suas interações [FAR 96].


Segundo Uschold e Gruninger [USC 96] ontologia é uma reunião explícita de conhecimento compartilhado em uma dada área, conseqüentemente pode resolver problemas de comunicação entre pessoas, organizações e sistemas de software, assim uma ontologia pode funcionar como um *framework* de unificação entre diferentes pontos de vista.

Gruber [GRU 93] diz que, uma ontologia é “uma especificação explícita e formal de uma conceituação compartilhada”, onde, explícita significa que o tipo de conceito

utilizado e as restrições sobre eles são explicitamente definidos, formal refere-se ao fato de que a ontologia pode ser lida por máquina (*machine readable*) e conceituação refere-se ao modelo abstrato do fenômeno que teve seus conceitos relevantes identificados.

A tabela 2.1, a seguir, apresenta um dado **domínio** (medicina), dois de seus **conceitos** fundamentais (doença e sintoma), uma das **relações** possíveis entre os conceitos apresentados (causa) e ainda uma das **restrições** que devem ser observadas entre seus conceitos (que uma doença não causa a si própria, querendo-se dizer com isso que uma doença causa um ou mais sintomas mas que nunca uma doença pode se auto causar).

TABELA 2.1 - Exemplos de conceitos utilizados em ontologias.

Domínios	Conceitos = Classe	Relações	Restrições
Medicina	Doença		Uma doença não causa a si própria
	Sintoma		

Dependendo de seu nível de generalidade, diferentes tipos de ontologias que cumprem diferentes papéis, podem ser identificadas [STU 99], estes papéis incluem principalmente os seguintes tipos: (a) Ontologias de domínio que capturam o conhecimento válido para um tipo particular de domínio; (b) Ontologias genéricas ou de senso comum que capturam conhecimentos gerais sobre o mundo e proporcionam noções e conceitos básicos sobre coisas como, tempo, espaço, estado, etc.; (c) Ontologias representacionais que não se prendem a nenhum domínio em particular, permitem representar entidades sem declarar o que deve ser representado. Exemplo deste tipo de ontologia pode ser visto em Gruber [GRU 93]; (d) Ontologias de tarefa que proporcionam termos específicos para tarefas em particular, por exemplo, hipótese pertence a ontologia de tarefa conhecida como diagnóstico; (e) Ontologias de método que proporcionam termos específicos para métodos de resolução de problemas .

Uma ontologia pode assumir diversos formatos [USC 96], mas deve incluir necessariamente um vocabulário de termos e suas definições, isto é, alguma especificação quanto a seus significados. Segundo Uschold e Gruninger [USC 96] o grau de formalismo pode assumir quatro níveis distintos, a saber: (a) altamente informal, em linguagem natural, (b) semi-informal, usando uma forma restrita e estruturada da linguagem natural mais clara e menos ambígua, (c) semiformal, usando uma linguagem artificial formalmente definida e (d) rigorosamente formal.

Ontologias podem ser usadas para resolver problemas de: (a) Comunicação entre pessoas com diferentes necessidades e diferentes pontos de vista, existentes em função de seus diferentes contextos; (b) Interoperabilidade entre sistemas de software o que é obtido pela tradução entre diferentes métodos de modelagem, paradigmas, linguagens e ferramentas de software; (c) Engenharia de software como reusabilidade, confiabilidade e de especificação [USC 96].

Se uma ontologia é um *framework* para comunicação entre pessoas sua, representação pode ser informal, desde que seja precisa. Já, se a ontologia for ser usada por um software (*readable machine*), então sua semântica deve ser muito mais precisa. Ontologias podem, também, assumir o papel de padronizar representações entre ferramentas, permitindo que elas interoperem e compartilhem uma compreensão [USC 96].

Ontologias são implementadas segundo objetivos específicos [USC 96]. No projeto Plinius, por exemplo o objetivo é a extração semi-automática de conhecimento a partir

de textos em linguagem natural, mais especificamente sobre propriedades mecânicas de materiais cerâmicos. Neste projeto, um léxico é usado para mapear a linguagem natural para expressões formais em uma linguagem de representação de conhecimento e a ontologia específica a linguagem na qual a parte semântica do léxico é expressa.

2.2.1 Visão Geral Sobre Methontology

Methontology é uma metodologia de desenvolvimento de ontologias que tem estado em desenvolvimento na Universidad Politécnica de Madrid. A principal virtude dos artigos relativos a esta metodologia reside no fato de, ao contrário da grande maioria dos artigos que descrevem metodologias, técnicas ou ferramentas, este descreve detalhadamente, além da metodologia propriamente dita, como a informação produzida por cada etapa da metodologia é mantida computacionalmente, apresentando as tabelas relacionais usadas para este fim. Este tipo de informação foi extremamente útil para a elaboração do modelo de dados para a proposta alvo deste trabalho e que está descrita no capítulo 3, assim os principais componentes presentes na descrição de Methontology e que tiveram especial influência na elaboração deste trabalho são citadas a seguir. Diversas outras figuras, tabelas e descrições da metodologia merecem ser analisados, mas foram propositalmente omitidos, podendo ser encontrados em Blázquez et al [BLA 98], Fernández, Gómez-Péres e Juristo [FER 97] e Gómez-Péres, Fernández e Vicente [GOM 96] em detalhe ou em Santi [SAN 00] resumidamente.

Methontology constitui-se de um processo em diversas etapas para a construção de ontologias de domínio, onde, ao final de cada etapa, um resultado é esperado e é este resultado que dá nome a etapa. Cada uma destas etapas foi implementada compondo uma ferramenta chamada ODE (Ontology Development Environment) [GOM 96] e serão sucintamente descritas a seguir:

- **Glossário de termos** – descreve os termos (conceitos, instâncias, atributos, verbos, etc.) do domínio.
- **Árvore de classificação de conceitos** – são estruturas auxiliares quando o número de conceitos é significativo (o que normalmente ocorre). Os conceitos são organizados em árvores que são construídas pelo uso de relações de classe e que permitem identificar as taxonomias do domínio.
- **Diagrama de relações binárias** – identifica graficamente cada uma das relações binárias existentes entre os conceitos identificados nas etapas anteriores.
- **Dicionário de conceitos** – relaciona para cada conceito seus sinônimos, acrônimos, exemplos de instâncias, atributos da classe e da instância e as relações binárias das quais o conceito participa.
- **Tabela de relações binárias** – relaciona os atributos necessários a representação das relações binárias identificadas, conforme exemplo abaixo:

TABELA 2.2.a - Relação binária emprega.

Nome da relação	Emprega
Conceito fonte	Organização
Cardinalidade fonte	(1,n)
Conceito alvo	Funcionário
Propriedades matemáticas	
Relação inversa	Afiliação
Referências	

TABELA 2.2.b - Relação binária afiliado.

Nome da relação	Afiliação
Conceito fonte	Funcionário
Cardinalidade fonte	(1,n)
Conceito alvo	Organização
Propriedades matemáticas	
Relação inversa	Emprega
Referências	

- **Tabela de atributos de instâncias** – relaciona, para cada instância, os atributos necessários a representação das instâncias identificadas, conforme exemplo a seguir:

TABELA 2.3.a - Atributos da instância fotografia.

Nome do atributo de instância	Fotografia
Tipo de valor	String
Unidade de medida	
Precisão	
Intervalo de valores	
Valor padrão	
Cardinalidade	(1,N)
Inferido do atributo de instância	
Inferido do atributo de classe	
Inferido de constante	
Fórmula	
Inferir	Beleza
Referências	

TABELA 2.3.b - Atributos da instância peso.

Nome do atributo de instância	Peso
Tipo de valor	Quantidade
Unidade de medida	Quilograma
Precisão	0,001
Intervalo de valores	[0,200]
Valor padrão	
Cardinalidade	(1,1)
Inferido do atributo de instância	
Inferido do atributo de classe	
Inferido de constante	
Fórmula	
Inferir	
Referências	

- **Tabela de atributos de Classe** - para cada atributo de classe deverá ser fornecido: o nome, tipos de valores possíveis, unidades de medida para valores numéricos, precisão, atributos de instâncias que podem ser deduzidos usando o valor deste atributo e referências.
- **Tabela de axiomas lógicos** - São usadas para definir conceitos pelo significado de expressões lógicas que são sempre verdadeiras.

TABELA 2.4.a - Axioma estudante de PhD ...

Nome do axioma	Estudante-de-PhD-não-esta-em-banca-de-doutorado
Descrição	Um estudante de PhD não pode estar em uma banca de doutorado
Conceito	Estudante de PhD
Atributos refer	
Variáveis	S P
Expressão	ParaTodo (S,P) EstudantePhD (S) => not (MembroBancaDoutorado (S,P))
Relações	
Referências	

TABELA 2.4.b - Axioma líder de projeto ...

Nome axioma	Lider-de-Projeto-Trabalha-no-Projeto
Descrição	O funcionário que lidera o projeto, também trabalha no projeto
Conceito	Funcionário
Atributos refer	
Variáveis	E P
Expressão	ParaTodo (E,P) Funcionário e Lider-de-projeto (E,P) => Trabalhar-no-projeto (E,P)
Relações	
Referências	

- **Tabela de constantes** – relaciona cada constante identificada no processo. Seus atributos são exemplificados a seguir:

TABELA 2.5 - Constantes do domínio publicações.

Nome constante	Descrição	Valor	Unidade	Inferir
Artigo-em-congresso	Coefficiente aplicado para obter o padrão de publicação de artigo em congresso	0,75		Taxa-de-publicação-padrão
Artigo-em-workshop	Coefficiente aplicado para obter o padrão de publicação de artigo em workshop	0,50		Taxa-de-publicação-padrão

- **Tabela de fórmulas** – relaciona cada fórmula identificada no processo. Seus atributos são exemplificados abaixo:

TABELA 2.6 - Fórmula para normalização de publicações.

Nome da fórmula	Fórmula-para-taxa-padrão-de-publicação
Conceito	Pessoa
Atributo inferido	Taxa-padrão-de-publicação
Fórmula	Taxa-padrão-de-publicação = Artigos-em-jornal + Constante-artigos-em-congresso * Artigos-em-congresso + Constante-artigos-em-workshop * Artigos-em-workshop
Descrição	Para padronização de publicações um artigo em jornal tem valor 1, artigos em congresso 0,75 e em workshop 0,5
Atributos básicos	Artigos-em-jornal, artigos-em-congresso, artigos-em-workshop
Constantes	Constante-artigo-em-congresso, constante-artigo-em-workshop
Precisão	0,01
Restrições	
Referências	

- **Árvore de classificação de atributos** – relaciona conceitos complexos e seus conceitos formadores. Graficamente o conceito complexo é mostrado como elemento raiz da árvore e seus conceitos formadores como folhas. No exemplo acima Taxa_Padrão_de_Publicação é o conceito complexo e artigos em congresso, workshop, jornal, constante artigos em jornal e constante artigos em workshop seus componentes.
- **Tabela de instâncias** – relaciona os atributos necessários a manutenção das instâncias.

TABELA 2.7 - Instâncias do domínio.

Instância	Atributo	Valor
Pedro Alvarez Cabral	Nome completo	“Pedro Alvarez Cabral”
	Primeiro Nome	“Pedro”
	Último Nome	“Cabral”
	E-Mail	“pcabral@naucapitanea.com.pt”

2.2.2 Aplicações de Ontologias a Dados Semi-estruturados

Esta seção tem o objetivo de apresentar algumas iniciativas de integração de dados semi-estruturados através do uso de ontologias, bem como apresentar trabalhos que influenciaram esta proposta. As informações resumidamente descritas a seguir foram extraídas dos trabalhos de Mello e Heuser [MEL 01], Observer, Ontobroker e Momis e de Santi [SAN 00], On2broker e Infomaster. Sugere-se a leitura destes trabalhos bem como dos trabalhos neles citados para uma descrição detalhada destas iniciativas. A seguir serão descritas, sucintamente, algumas iniciativas de uso de ontologias para integração de fontes heterogêneas e distribuídas de informação semi-estruturada.

- **Observer** é um sistema de informação global, em desenvolvimento na Universidade Politécnica de Madrid, que utiliza ontologias para o processamento de consultas a fontes de dados heterogêneos estruturados e semi-estruturados. Ontologias descrevem conceitos e regras organizados em uma hierarquia de especializações que captura a semântica do conteúdo de uma ou mais fontes. Dois tipos de relacionamentos existem neste contexto: (i) relacionamento entre ontologias, onde cada conceito de uma ontologia pode ter uma informação de mapeamento para conceitos da mesma ou de outras ontologias. Os tipos de relacionamentos são: (a) sinônimo, (b) mais geral, (c) menos geral, (d) interseção, (e) disjunção e (f) cobertura; (ii) relacionamento entre ontologia e fonte de dados, onde cada conceito da ontologia possui uma informação de mapeamento que a relaciona com estruturas das fontes de dados através de uma álgebra relacional estendida. Uma fonte é alcançada por apenas uma ontologia.
- **Ontobroker** é uma ferramenta baseada em ontologias da Universidade de Karlsruhe, Alemanha, que processa documentos especificados em linguagens de marcação como HTML e XML, provendo recuperação inteligente de informação. Ontobroker possui uma arquitetura formada por um mecanismo de consulta, um agente de informação e uma máquina de inferências. As ontologias são o princípio geral da estruturação de dados: o agente as utiliza para extrair fatos, o mecanismo de inferências para estruturar dados e o mecanismo de consulta para formular as consultas. No Ontobroker especificações ontológicas são usadas para gerar DTDs XML que por sua vez definem classes de conceitos que auxiliarão na marcação de documentos XML. A DTD não impõe restrições sobre qual dos elementos definidos deve ser a raiz no documento XML, nem quanto a existência de todos os elementos.
- **On2Broker** é uma atualização do projeto Ontobroker, que faz por assim dizer um serviço de corretagem (*brokering*) para melhorar o acesso a fontes de informação heterogênea, distribuída e semi-estruturada, como as disponíveis na WWW. Para isso processa fontes

de informação e descrições de conteúdo em HTML, XML, e RDF e proporciona recuperação de informações, resposta a consultas e suporte a manutenção. On2broker utiliza ontologias, como forma de explicitar a semântica das páginas *Web* e para descrever conhecimento. Sua arquitetura compõe-se de: (i) um agente de informação que extrai fatos das fontes, (ii) um mecanismo de inferência, (iii) um SGBD e (iiii) um mecanismo de consultas. Todos estes componentes utilizam ontologias para desempenhar suas funcionalidades. Para formular as ontologias é utilizada uma linguagem baseada na lógica de *frames* que fornece classes, atributos com definição de intervalo e domínio, hierarquias *is-a* com subclasses e herança e axiomas lógicos

- **Momis** (*Mediator environment for Multiple Information Sources*) é um ambiente para integração de fontes de dados estruturados e semi-estruturados, em desenvolvimento nas universidades de Modena e Milão na Itália. Sua arquitetura tem como componentes: (i) um esquema conceitual ou ontológico orientado a objetos especificado na linguagem ODL (linguagem de integração semântica); (ii) *wrappers* para tradução de esquemas em representações ODL e; (iii) componentes mediador e processador de consultas. Ontologias são descritas através de três tipos de relacionamentos entre classes e atributos: (i) sinônimo (t1 SYN t2), (ii) mais_geral (t1 BT t2) e relacionado (t1 RT t2). O relacionamento inverso a mais_geral é o relacionamento mais_específico e o processo de determinar todas estas relações é um processo semi-automático. O mediador utiliza a ontologia para gerar um esquema global que recebe coeficientes de afinidade entre conceitos de fontes distintas. Para o esquema global, no caso de sinônimos apenas um elemento é gerado e no caso das relações mais_geral e relacionado, apenas o termo de significado mais amplo é selecionado.
- **Infomaster** é um sistema de integração de informações que oferece acesso integrado a múltiplas fontes de informação heterogênea e distribuída sobre a internet, passando uma impressão de um sistema de informação homogêneo e centralizado, isto é, um *datawarehouse* virtual. Segundo o Dr. Arthur M. Keller [KEL 00]: “o Infomaster usa ontologias para descrever os diversos modelos de dados das bases de dados a serem integradas, bem como, das visões de dados a serem apresentadas”.

2.3 Dados Semi-Estruturados

Dados semi-estruturados denotam um conjunto bastante grande de estruturas, que se situam entre dois extremos bastante distintos (figura 2.1). De um lado, dados que não possuem nenhuma estrutura pré-determinada (arquivos binários, por exemplo) e de outro aqueles fortemente estruturados, como os bancos de dados cujas informações devem estar de acordo com um esquema previamente definido [ABI 97]. Dentro do contexto de dados semi-estruturados encontram-se formatos como HTML (*Hiper-text Markup Language*), XML (*Extensible Markup Language*) e SGML (*Standard Generalized Markup Language*) [DOR 00]. A figura 2.1 mostra que os dados semi-estruturados situam-se entre os dois extremos citados acima (dados binários e BDs) e mostra também os diferentes graus de estruturação dos diferentes casos de dados semi-estruturados.

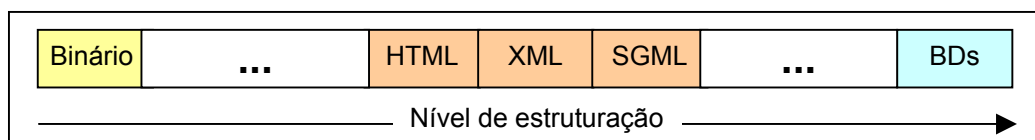


FIGURA 2.1 - Classificação das estruturas de dados quanto ao nível de estruturação.

Um documento semi-estruturado compõe-se, basicamente, de conceitos que aparecem de forma explícita ou não. Estes conceitos podem ser léxicos, se representarem a informação diretamente ou não léxicos (complexos) se forem compostos de outros conceitos léxicos ou complexos [DOR 00]. Estes conceitos, bem como as relações entre eles são o objeto de interesse deste trabalho, isto é, aquilo que se deseja descobrir e integrar.

A abordagem que será feita a seguir refere-se basicamente a XML, suas principais características e seus componentes, procurando deixar claro porque o enfoque deste trabalho se dá sobre XML e não por algum outro padrão. O principal motivo em seu uso deve-se ao fato de que XML é uma linguagem de marcação adequada a representação e ao intercâmbio de dados, documentos e outras entidades cuja essência é baseada na sua capacidade de unir informação. Outro fato que influenciou a escolha é seu elevado nível de padronização e sua aceitação em larga escala tanto nos meios acadêmicos como na indústria [MEL 02].

2.3.1 Introdução ao XML

O XML não foi originalmente concebido como uma tecnologia de banco de dados. De fato, como o HTML, no qual a *World Wide Web* é baseada, XML tem suas raízes no gerenciamento de documentos, e é derivado de uma linguagem para estruturação de grandes documentos conhecida como SGML. Porém, diferentemente de SGML e HTML, XML pode representar dados de um banco de dados, como também muitos outros tipos de dados estruturados usados em aplicações de negócios. O XML é particularmente útil como formato de dados quando uma aplicação precisa se comunicar com outra, ou para integrar informação de diversas aplicações. Quando XML é usado nestes contextos, muitos assuntos de bancos de dados tornam-se pertinentes, incluindo como organizar, manipular e consultar dados XML.

Para entender XML [SIL 02], é importante compreender suas raízes como uma linguagem de marcação de documentos. O termo marcação refere-se a qualquer coisa num documento que não é apresentada como saída impressa. Por exemplo, um escritor criando um texto que irá eventualmente ser composto como uma revista pode querer fazer notas sobre como a composição deve ser feita. Pode ser importante colocar estas notas de uma forma que elas possam ser distinguidas do conteúdo atual, assim uma nota como **não quebre este parágrafo** não será impressa na revista. No processamento eletrônico de documentos, uma linguagem de marcação é uma descrição formal de que parte do documento é conteúdo, que parte é marcação e o que as marcações significam.

Tal como os sistemas de bancos de dados evoluíram do processamento de arquivos físicos para oferecer visões lógicas distintas, as linguagens de marcação evoluíram de expressões que especificavam como imprimir as partes de um documento, para especificar a função de seu conteúdo, o que auxilia na extração de partes chave de um documento. HTML, SGML e XML usam *tags* entre os sinais de maior e menor para identificar a função da porção de texto envolvida por estas marcas, assim para indicar que o título de um documento é dados semi-estruturados pode-se usar a seguinte forma <título>Dados Semi-Estruturados</título>. Diferentemente de HTML, o XML não define o conjunto de *tags* permitidas, que pode ser estendido conforme necessário. Esta característica é que faz do XML adequado a representação e troca de dados. Da mesma forma como SQL é a linguagem predominante para consultas a dados relacionais, XML está se tornando o formato dominante para intercâmbio de dados.

2.3.2 A Estrutura dos Dados XML

A construção fundamental em um documento XML é o *element*. Um elemento é simplesmente um par de *tags* de início e fim e todo o texto que aparece entre eles. Documentos XML têm um único elemento **raiz** que envolve todos os outros elementos do documento, isto é, todos os demais elementos devem estar aninhados ao elemento raiz. Além dos elementos, há também a noção de **atributo** que aparece como um par nome=valor antes da marca de fechamento (>). Como documentos XML foram projetados para ser trocados entre aplicações, um mecanismo de *name-space* que permita especificar globalmente nomes únicos, como endereços *Web*, por exemplo, foi incluído na especificação XML.

2.3.3 Esquemas para Documentos XML

Bancos de dados tem esquemas, que são usados para restringir que informações podem ser armazenadas no BD e para restringir os tipos de dados da informação armazenada. Já documentos XML, por padrão, não precisam ser criados tendo um esquema associado a eles. Esta liberdade é aceitável quando não envolve processamento automático de documentos. A seguir serão descritos dois mecanismos de esquema orientado a documentos que o padrão XML define, a *Document Type Definition (DTD)* e *XMLSchema*.

A DTD é uma parte opcional de um documento XML e seu principal propósito é servir como uma lista de regras para padrões de aninhamento entre sub-elementos e elementos. Cada declaração pode conter operadores que permitam determinar a **cardinalidade** da relação entre um elemento e seus sub-elementos, estes operadores podem ser uma **barra** vertical (|) que indica a opcionalidade entre os elementos separados por ela, um sinal de **adição** (+) que indica que o sub-elemento pode ocorrer uma ou mais vezes, o sinal **asterisco** (*) que indica zero ou mais ocorrências e finalmente o ponto de **interrogação** (?) que indica zero ou uma ocorrência do sub-elemento.

Elementos de uma DTD podem ser definidos como sendo do tipo *#PCDATA*, *empty* ou *any* significando respectivamente conter dados texto, não conter dados e qualquer elemento não mencionado na DTD podendo ocorrer. Cada elemento pode ter atributos dos tipos *CDATA* que informa que o atributo contém dados do tipo caracter, *ID* que provê um identificador único para o elemento, *IDREF* que é uma referência a um elemento do documento e precisa conter um valor que apareça no atributo *ID* de algum elemento ou *IDREFS* que permite uma lista de referências.

Já *XMLSchema* representa um esforço para corrigir muitas das deficiências das DTDs. Alguns destes acréscimos serão descritos a seguir sem a pretensão de cobrir totalmente a sintaxe do *XMLSchema*. No *XMLSchema* é possível especificar o tipo de um atributo através do uso de estruturas como *xsd:string* ou *xsd:decimal*, é possível definir o número mínimo e máximo de ocorrências de um sub-elemento através de *minOccurs* e *maxOccurs*, é possível criar tipos complexos usando uma forma de herança, é possível criar tipos definidos pelo usuário e é possível impor restrições de unicidade e chave estrangeira a um atributo. Torna-se claro que *XMLSchema* acrescenta características desejáveis ao processo de mapeamento (descrito no capítulo 3) em relação as DTDs. Em contra-partida eles são significativamente mais complexos que as DTDs e não serão considerados neste trabalho, no momento.

2.3.4 A Interface de Programação de Aplicações (API)

Com a ampla aceitação de XML como um formato de representação e intercâmbio de dados, um grande número de ferramentas de software estão disponíveis para manipulação de dados XML. De fato, existem dois modelos padrão para manipulação de XML através de programas, cada um deles disponíveis para uso com uma grande variedade de linguagens de programação.

Uma das APIs padrão para manipulação de XML é o *document object model* (DOM), que trata o conteúdo XML como árvores, com cada elemento representado por um nodo, chamado *DOMNode*. Programas podem acessar partes do documento de uma forma navegacional, iniciando pela raiz. Bibliotecas DOM são disponíveis para muitas linguagens de programação e estão presentes em *Web browsers*, onde ele pode ser usado para manipular o documento apresentado ao usuário. A seguir serão apresentadas algumas das interfaces e métodos na API Java para DOM. A API Java para DOM proporciona uma interface chamada *Node* e as interfaces *Element* e *Attribute*, que herdaram da interface *Node*. A interface *Node* oferece métodos como *getParentNode()*, *getFirstChild()* e *getNextSibling()*, para navegar pela árvore DOM, iniciando pelo nodo raiz. Subelementos de um elemento podem ser acessados pelo nome *getElementsByTagName(name)*, que retorna uma lista de todos os elementos filhos com um específico nome de tag; membros individuais de uma lista podem ser acessados pelo método *item(i)*, que retorna o *i*-ésimo elemento de uma lista. Os valores dos atributos de um elemento podem ser acessados por nome, usando o método *getAttribute(name)*. O texto contendo o valor de um elemento é modelado como um nodo *Text*, que é filho de um elemento nodo. O método *getData()* do nodo *Text* retorna seu conteúdo. DOM também proporciona uma variedade de funções para atualizar o documento adicionando e excluindo atributos e elementos filhos de um nodo, atribuindo valores ao nodo e etc.

A segunda interface de programação que será discutida aqui é a *Simple API* para XML (SAX), que é um modelo de eventos, projetado para oferecer uma interface comum entre *parsers* e aplicações. Esta API é construída sobre a noção de manipuladores de eventos (*event handlers*), que consistem de funções especificadas pelo usuário associadas a análise de eventos (*parsing events*). Análise de eventos corresponde ao reconhecimento de partes de um documento, por exemplo, um evento é gerado quando uma *start-tag* é encontrada para um elemento, e outro evento é gerado quando um *end-tag* é encontrado. As partes de um documento são sempre encontradas na ordem de início a fim. SAX não é adequado para aplicações de bancos de dados.

2.4 Síntese do Capítulo

Neste capítulo foi apresentada uma síntese dos trabalhos mais recentes e significativos identificados em pesquisa realizada procurando mostrar o estado da arte em termos dos principais componentes que subsidiam este trabalho e que são: (1) integração de esquemas de bancos de dados tradicionais, (2) ontologias e (3) dados semi-estruturados.

Com o tópico integração de esquemas procurou-se identificar na literatura pontos que permitissem inferir, por analogia, um método de integrar esquemas para dados semi-estruturados. Assim enquanto, tradicionalmente, integração de esquemas objetiva integrar esquemas de diferentes BDs compostos de uma série de inter-relações de conceitos, neste trabalho deseja-se integrar esquemas representativos de conjuntos de documentos XML que por sua vez contém as inter-relações de conceitos presentes nestes documentos. Para isto,

na seção destinada a integração de dados (seção 2.1), foram analisados trabalhos referentes a diferentes topologias que tratam de integração de bases de dados distribuídas e heterogêneas:

(i) integração global de esquemas, (ii) bancos de dados federados e (iii) linguagens para múltiplos bancos de dados [BOU 99]. Foram analisadas as causas da diversidade representacional [ELM 99 e BAT 86] e uma taxonomia de conflitos de representação [SHE 90 e BAT 86], bem como, formas de resolver esta heterogeneidade semântica [HAM 99 e BAT 86]. Também foram estudadas estruturas de integração, técnicas de identificação de relações e de geração do esquema integrado, métodos de integração e ferramentas que automatizam parcialmente o processo [RAM 99]. Destas análises foram tiradas as soluções adotadas na proposta objeto deste trabalho.

Na seção destinada a ontologias foi feito um estudo das origens do termo ontologia, para que servem [USC 96] e seus tipos. Uma metodologia de criação de ontologias de domínio chamada Methontology [FER 97] foi especialmente estudada. Dela derivou-se a estrutura básica de que entidades seriam necessárias à manutenção dos dados necessários para esta proposta. Também foram analisadas cinco iniciativas de integração de informações feitas com base em ontologias [DEC 98, DUS 97, FEN 99, MEL 00 e SAN 00].

Finalmente, o capítulo sobre integração de dados discorre sobre dados semi-estruturados [DOR 00] em geral e sobre dados XML em específico [PIM 00 e SIL 02]. Nesta seção é feita uma rápida introdução aos dados semi-estruturados e uma descrição relativa a documentos XML, sua estrutura, os esquemas utilizados para representá-los e finalmente foi feita uma pequena descrição das interfaces de programação de aplicações (APIs) disponíveis para XML [SIL 02] e que foram utilizadas na implementação de um protótipo [MAR 00] que é descrito no anexo 1, ao final deste trabalho.

3 A Proposta de Mapeamento e Integração

O enorme volume de informação publicada na *Web*, na forma de documentos semi-estruturados, tem estimulado um grande número de pesquisas com temas que vão da extração [BER 99 e DOR 00] à integração de informação semi-estruturada [DEC 98, DUS 97, FEN 99 e GEN 97] entre outros. Esta proposta visa integrar esquemas representativos de informação semi-estruturada, ou em específico integrar os conceitos presentes em DTDs XML, com o objetivo de oferecer um modelo dos conceitos envolvidos em um dado domínio que teve seus esquemas (DTDs) integrados.

A origem desta proposta parte da necessidade, citada no parágrafo anterior, de integrar semanticamente informação semi-estruturada, disponível na *Web*. Diz-se semântica no intuito de agregar mais significado ao modelo através da integração e em contraposição as integrações meramente léxicas comumente encontradas em mecanismos de busca. O conteúdo da *Web*, compõe-se basicamente de páginas HTML e, mais recentemente, documentos XML. A larga aceitação que os documentos XML tem tido entre usuários, provedores de conteúdo e fornecedores de software, entre outros, bem como sua padronização, colocam este formato como um excelente candidato à fonte de informação a integrar. Aliado a existência de uma padronização consistente e largamente aceita existe o fato de que documentos XML normalmente estão associados a esquemas (DTD ou XMLSchema) que definem sua estrutura interna. Uma DTD-XML organiza seus conceitos em uma estrutura de árvore e contém símbolos que permitem determinar parcialmente as cardinalidades das relações entre seus conceitos (nodos da árvore). Uma DTD visualizada como uma árvore de conceitos inter-relacionados, acrescida dos rótulos e das cardinalidades destes relacionamentos proporciona um acréscimo significativo na semântica da informação comunicada ao usuário. Conseqüentemente, a integração de diversas árvores de conceitos agregará ainda mais semântica ao modelo conceitual gerado³.

A seção 2.1 deste trabalho preconiza a utilização de um modelo comum de dados para o qual devem ser convertidos os modelos de dados heterogêneos que se deseja integrar. Como o objetivo desta proposta é integrar formatos que vão de dados semi-estruturados a dados estruturados e como muitos dos esquemas para estes tipos dados não permitem representar toda a semântica pretendida (nomes de relacionamentos e cardinalidades entre outros) é necessária a adoção de um modelo comum. Em função destas necessidades adotou-se o modelo E-R, por que ele é utilizado em integração de esquemas de BDs heterogêneos [RAM 99 e BON 94], porque é capaz de manter adequadamente tanto as estruturas na forma de árvores do processo de mapeamento, quanto as estruturas na forma de grafos do processo de integração e porque é capaz de oferecer suporte adequado quando o usuário deseja incluir novas informações ao modelo.

Este modelo integrado de conceitos relacionados foi chamado de ontologia, termo este que tem sido largamente utilizado em diversos trabalhos (conforme apresentado na seção 2.2). No livro *Management of Heterogeneous and Autonomous Database Systems*, em seu capítulo 4, Hammer e McLeod apud Fang et al [FAN 91] utilizam, em sua arquitetura de integração, um léxico local que se constitui de fatos que representam conhecimento, dizem também que, a lista de conceitos consensuais é chamada de ontologia e é distinta para domínios de aplicação igualmente distintos. Pesquisadores da área de IA como, Uschold e Gruninger [USC 96], dizem que “uma ontologia pode assumir diversos formatos, mas deve incluir um vocabulário de termos e suas definições”. Em outro ponto do mesmo trabalho dizem: “ontologia é uma especificação explícita e formal de uma conceituação

³ Este modelo conceitual passará a ser chamado de ontologia, conforme explicado no próximo parágrafo.

compartilhada”. Todas estas descrições combinam perfeitamente com os esquemas integrados que esta proposta tem como objetivo gerar. Se este modelo integrado de conceitos constitui uma ontologia é conveniente, para fins de diferenciação, que cada uma das DTDs (árvores de conceitos) que serão integradas para dar origem a esta ontologia seja chamada pré-ontologia. O termo pré-ontologia foi cunhado neste trabalho porque cada uma das árvores, se tomadas individualmente, não constitui uma ontologia. Mas é a partir da integração da primeira destas pré-ontologias que a ontologia passa a existir.

Os conflitos semânticos que ocorrem durante o processo de integração das pré-ontologias sobre a ontologia são tratados através de técnicas largamente aceites de integração de esquemas de BDs, conforme foi descrito na seção 2.1, onde esquemas ou visões individuais são integradas como forma de integrar os diversos BDs envolvidos.

Este capítulo abrange os seguintes tópicos. A seção 3.1 faz uma breve introdução onde descreve o processo de mapeamento e integração como um todo. As convenções utilizadas e as representações visuais adotadas nesta proposta são descritas na seção 3.2. Em seguida, na seção 3.3, o meta-modelo de mapeamento e integração é apresentado e discutido, descrevendo cada uma de suas entidades, atributos e relacionamentos, bem como a estrutura proposta para o *thesaurus*. Os algoritmos utilizados no processo de mapeamento e de integração são apresentados, descritos e comentados na seção 3.4, nela também é descrita a forma como são resolvidos os conflitos de nomenclatura, de cardinalidade e a forma como são analisadas a possibilidade de inserção de estruturas generalização/especialização. A seção 3.5, apresenta exemplos de muitos dos conflitos clássicos da literatura de integração de BDs [BAT 86] e como são tratados nesta proposta de integração de esquemas de dados semi-estruturados. A seção 3.6 descreve as intervenções e revisões realizadas pelo usuário (processo semi-automático) nas cardinalidades, mapeamentos e integrações. Na seção 3.7 é feito um relato das influências que este trabalho sofreu, com base na análise realizada na literatura sobre trabalhos relacionados (capítulo 2), e que culminou nesta proposta. Finalmente, na seção 3.8, é apresentada uma síntese deste capítulo.

3.1 Visão Geral do Proposta

O processo de integração proposto compõe-se de duas partes principais (figura 3.1). Uma **mapeia** (*wrapper*) todas as DTDs-XML selecionadas pelo usuário para pré-ontologias e outra **integra** cada pré-ontologia sobre a ontologia base. A base de ontologias, antes do primeiro mapeamento ocorrer, estará vazia. Após o primeiro mapeamento, o conteúdo da base de ontologias será a pré-ontologia utilizada no primeiro mapeamento, só que em um formato adequado ao processo de integração (modelo comum de integração). Os dois repositórios apresentados na figura 3.1, para pré-ontologias e para ontologias, indicam uma separação lógica. Na prática, contudo, esta separação não existe até porque pré-ontologias e ontologias são uma especialização de esquema, como pode ser verificado na figura 3.6 que mostra o meta-modelo de integração.

Para ilustrar o processo, considera-se uma fonte de dados com n DTDs a integrar e os repositórios de pré-ontologias e de ontologias vazios. O processo se dará em duas fases conforme descrito a seguir:

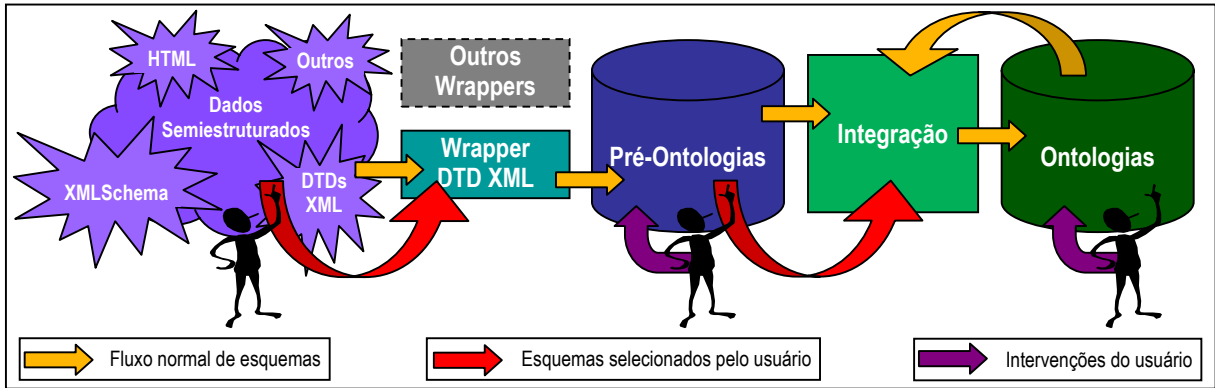


FIGURA 3.1 - Diagrama da proposta de integração de fontes de informação semi-estruturada.

- Na primeira fase, figura 3.2, os n esquemas de interesse para o usuário serão **mapeados**, um a um, para pré-ontologias, gerando assim, uma pré-ontologia para cada DTD de interesse. A qualquer momento, após o mapeamento, o usuário pode revisar nomes de conceitos, registrar os nomes de relacionamentos e revisar cardinalidades;
- Na segunda fase (figura 3.2), os esquemas serão **integrados** e após a primeira integração, a ontologia será constituída pela primeira pré-ontologia utilizada no processo de integração, uma vez que ainda não existe uma ontologia no repositório. A partir deste ponto, toda nova pré-ontologia passa a ser integrada sobre uma ontologia pré-existente. Este processo ocorrerá tantas vezes quantas forem as pré-ontologias a integrar. Após a integração o usuário pode revisar nomes de conceitos, de relacionamentos, cardinalidades e especializações.

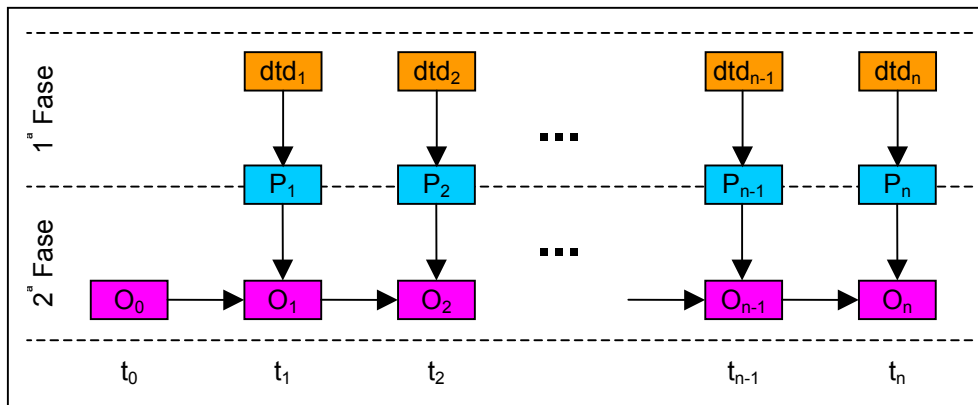


FIGURA 3.2 - 1ª fase – o processo de mapeamento das DTDs em pré-ontologias e 2ª Fase – o processo de integração das pré-ontologias sobre as ontologias.

A principal função do mapeamento (primeira fase) é homogeneizar a forma de representação. Assim, após o mapeamento, pré-ontologias geradas a partir de DTDs-XML e ontologias encontram-se representadas da mesma forma. No futuro quando forem criados mapeamentos para outras fontes de dados semi-estruturados, estas também serão mapeadas para o modelo comum de integração.

Na figura 3.2 é apresentada a característica incremental do processo de integração. Em um momento inicial t_0 a base de ontologias apresenta-se vazia. A partir deste momento o processo de integração é realizado integrando uma pré-ontologia previamente mapeada P_1 sobre a ontologia existente O_0 . Assim O_1 representa uma ontologia resultante da integração da pré-ontologia P_1 sobre a ontologia existente e vazia O_0 . A seguir a pré-ontologia P_2 , será integrada sobre a ontologia O_1 dando origem, através do processo de integração, a ontologia

O₂. Este processo se repetirá até que todas as pré-ontologias selecionadas pelo usuário tenham sido integradas.

3.2 Convenções e Representação Gráfica

Esta seção do trabalho tem o intuito de apresentar as convenções e representações visuais adotadas sem a preocupação de explicar o processo de mapeamento e integração, o que passa a ser feito nas seções seguintes. Para tal será apresentado, na figura 3.3, duas pequenas DTDs (figuras 3.3.a e 3.3.b). A representação gráfica adotada para representar visualmente a inter-relação entre os conceitos destas DTDs é mostrado na figura 3.4. As duas pré-ontologias resultantes do mapeamento das duas DTDs são mostradas nas figuras 3.5.a e 3.5.b. A representação da ontologia resultante da integração das duas pré-ontologias é mostrada na figura 3.5.c.

As duas DTDs apresentadas na figura 3.3, a seguir, são propositalmente pequenas e possuem conceitos facilmente identificáveis como passíveis de integração, como os conceitos **para** e **destinatário**, por exemplo. A figura 3.4, mostra as DTDs memorando e carta, da figura 3.3, representadas graficamente. Esta representação foi incluída nesta seção apenas para mostrar a estrutura em árvore típica de uma DTD em uma forma comumente encontrada na literatura e que permite identificar visualmente os conceitos e seus relacionamentos mais facilmente do que analisando diretamente o texto da DTD.

<pre><!DOCTYPE Memorando [<!ELEMENT Memorando (Cabeçalho Texto) <!ELEMENT Cabeçalho (De Para+ Assunto?) <!ELEMENT Texto (#PCDATA) <!ELEMENT De (#PCDATA) <!ELEMENT Para (#PCDATA) <!ELEMENT Assunto (#PCDATA)]></pre>	<pre><!DOCTYPE Carta[<!ELEMENT Carta (Envelope Texto) <!ELEMENT Envelope (Remetente Destinatário) <!ELEMENT Texto (#PCDATA) <!ELEMENT Remetente (#PCDATA) <!ELEMENT Destinatário (#PCDATA)]></pre>
a)	b)

FIGURA 3.3 - DTD para um memorando (a) e para uma carta (b).

Nesta figura foram utilizadas elipses para representar os conceitos e linhas para mostrar as relações entre estes conceitos. Elipses amarelas indicam conceitos complexos, isto é, conceitos não léxicos obtidos a partir da composição de um ou mais conceitos léxicos ou complexos, enquanto que elipses verdes indicam conceitos **folha**, isto é, conceitos léxicos [DOR 00 e MEL 02]. O conceito **raiz** de cada DTD é apresentado através de uma elipse com a borda mais espessa que as dos demais conceitos filhos (sub-elementos). Nomes de relacionamentos não fazem parte da DTD e serão acrescentados oportunamente pelo usuário sobre a pré-ontologia (ou ainda sobre a ontologia), e as cardinalidades destes relacionamentos, apesar de poderem ser determinadas, foram propositalmente omitidas visto o caráter ilustrativo da figura neste momento.

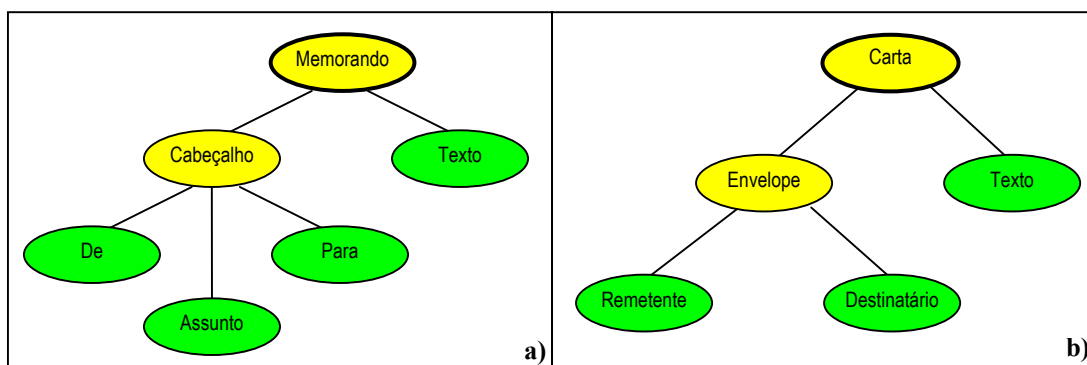


FIGURA 3.4 - A representação gráfica para a DTD de um memorando (a) e de uma carta (b).

A figura 3.5 mostra o resultado do mapeamento das DTDs memorando (figura 3.3.a) e carta (figura 3.3.b) para pré-ontologias. Para a representação das pré-ontologias e ontologias, foram utilizados retângulos para identificar os conceitos. Um retângulo, com as bordas mais largas que o normal, é utilizado para identificar o conceito raiz da pré-ontologia, já uma ontologia pode conter diversos conceitos raiz⁴ cada um deles referente a uma pré-ontologia integrada. As linhas que simbolizam as relações entre conceitos passam a ser nomeadas de **R1** até **Rn**, onde **n** representa o número de relações encontradas na DTD sendo mapeada. Estes nomes de relacionamentos correspondem a nomes internos e posteriormente ao mapeamento o usuário pode atribuir a eles nomes externos mais adequados. As cardinalidades das relações passam a ser expressas de forma completa⁵, se possível, e o usuário pode completar e/ou corrigir estas cardinalidades.

Quando uma cardinalidade não puder ser determinada de forma completa, o caracter ponto de interrogação (?) será utilizado para indicar esta incompletude. A figura 3.5.a mostra casos onde a cardinalidade não pode ser definida automaticamente com precisão e onde o caracter (?) foi utilizado para indicar esta situação. Na figura 3.5.b ocorre a mesma situação de dificuldade em precisar as cardinalidades completas, mas como as cardinalidades aparecem completas deduz-se que elas foram revisadas pelo usuário. Os nomes de conceitos, nomes de relacionamentos e cardinalidades que forem alteradas pelo usuário especialista serão identificadas por um losango (♦) a esquerda do rótulo deste conceito, relacionamento ou cardinalidade, como ocorre com o conceito raiz **Correspondência** e com o relacionamento **Informa** na figura 5.c. Sempre que dois conceitos, relacionamentos ou cardinalidades forem integrados, isto será explicitado sublinhando-se o rótulo identificador do objeto. Esta identificação de integração ou alteração é herdada, isto é, um objeto identificado como alterado ou integrado ao ser integrado a outro, levará consigo a marca e passará esta característica ao conceito resultante da integração. Isto pode ser observado, por exemplo, nas cardinalidades das relações R1 e R2 da figura 3.5.c, que teve o símbolo de alteração pelo usuário (♦) herdado das relações R1 e R2 da figura 3.5.b.

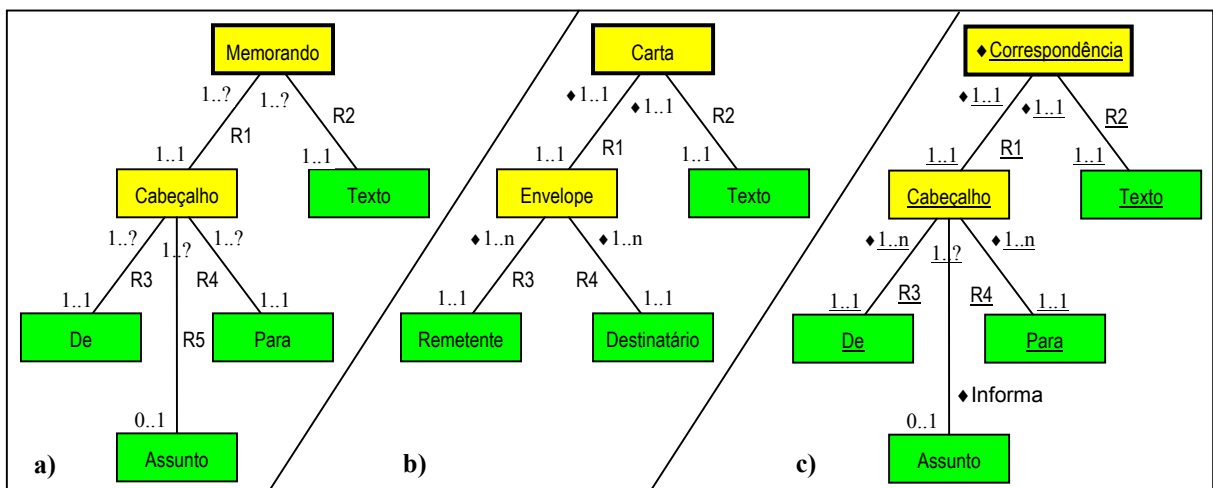


FIGURA 3.5 - Pré-ontologias das DTDs memorando (a) e carta (b) e a ontologia integrada resultante (c).

A tabela 3.1, abaixo, relaciona todos os construtores utilizados para representar graficamente a DTD, as pré-ontologias e as ontologias. Esta tabela identifica, também, o escopo de utilização destes construtores entre usuário especialista (aquele que participa ativamente do processo de mapeamento e integração) e usuário final (aquele que apenas

⁴ A rigor uma ontologia não possui um conceito raiz. As raízes que podem aparecer em uma ontologia tem origem no conceito raiz de cada pré-ontologia integrada.

⁵ A forma utilizada para determinar as cardinalidades é descrita na seção 3.4.4.

utiliza a ontologia resultante). As colunas usuário especialista e usuário final, referem-se ao que está acessível e/ou visível a estes perfis de usuário. Na tabela o caracter (P) indica que nomes de relacionamentos e cardinalidades podem estar visíveis ao usuário final se o usuário especialista identificá-las como públicas. Este artifício é utilizado porque o excesso de informação pode acabar poluindo a ontologia gerada, o que pode não ser conveniente.

TABELA 3.1 - Componentes visuais utilizados para representar pré-ontologias e ontologias.

Símbolo	Interpretação	DTD Gráfica	Pré-Ontologia	Ontologia	Usuário Especialista	Usuário Final
Elipse	Conceito	✓				
Retângulo	Conceito		✓	✓	✓	✓
Reta	Relacionamento	✓	✓	✓	✓	✓
Borda Normal	Conceito Normal	✓	✓	✓	✓	✓
Borda Espessa	Conceito Raiz	✓	✓	✓	✓	✓
Verde	Conceito Léxico	✓	✓	✓	✓	✓
Amarelo	Conceito Não Léxico	✓	✓	✓	✓	✓
Sublinhado	Resultante de Integração			✓	✓	
Losango	Resultante de Intervenção		✓	✓	✓	
Texto em Reta	Nome do relacionamento		✓	✓	✓	P
Cmax..Cmin	Cardinalidade máxima e mínima		✓	✓	✓	P
Triângulo	Generalização/especialização		✓	✓	✓	✓

Finalmente deve-se ressaltar que foi adotado como modelo conceitual, o modelo E-R, com algumas extensões que permitiram ampliar sua capacidade semântica tornando-o mais adequado e expressivo ao uso que se pretende dele. Estas extensões são a borda espessa para indicar um conceito raiz de uma pré-ontologia, as cores verde e amarelo para representar respectivamente conceitos léxicos e não léxicos, o uso de rótulos sublinhados para conceitos, relacionamentos e cardinalidades que foram integrados e por fim o uso de um losango a esquerda de um rótulo de conceito, relacionamento ou cardinalidade para indicar que sobre aquele objeto, houve uma intervenção por parte do usuário especialista.

3.3 O Meta-modelo de Integração

Na seção 2.2.1, que faz uma visão geral sobre Methontology, foi dito que uma de suas principais virtudes está na descrição de um modelo de dados capaz de manter as informações referentes a uma ontologia de domínio. Ao mesmo tempo percebeu-se que os trabalhos relativos a integração de esquemas de BDs, bem como, de integração de dados semi-estruturados não divulgam como as informações são estruturadas e mantidas. Junte-se a isto a abordagem pragmática pretendida neste trabalho e a intenção de propor um modelo capaz de suportar o processo proposto e obteve-se o meta-modelo apresentado na figura 3.6.

Uma pré-ontologia é a representação de uma DTD em um formato conveniente e que possa ser manipulado pelo usuário especialista e pelo processo de integração. O processo de integração, por sua vez, gera uma nova ontologia utilizando como entradas uma pré-ontologia e uma ontologia. Esta seção apresenta o meta-modelo capaz de armazenar pré-ontologias e ontologias (figura 3.6) descrevendo cada uma de suas entidades, atributos e relacionamentos. Também será apresentado um exemplo de DTD (figura 3.7.a), sua representação gráfica (figura 3.7.b), a pré-ontologia (figura 3.7.c) e a ontologia resultante do

processo de integração (figura 3.7.d). Finalmente será mostrada uma abstração quanto conteúdo das entidades do modelo.

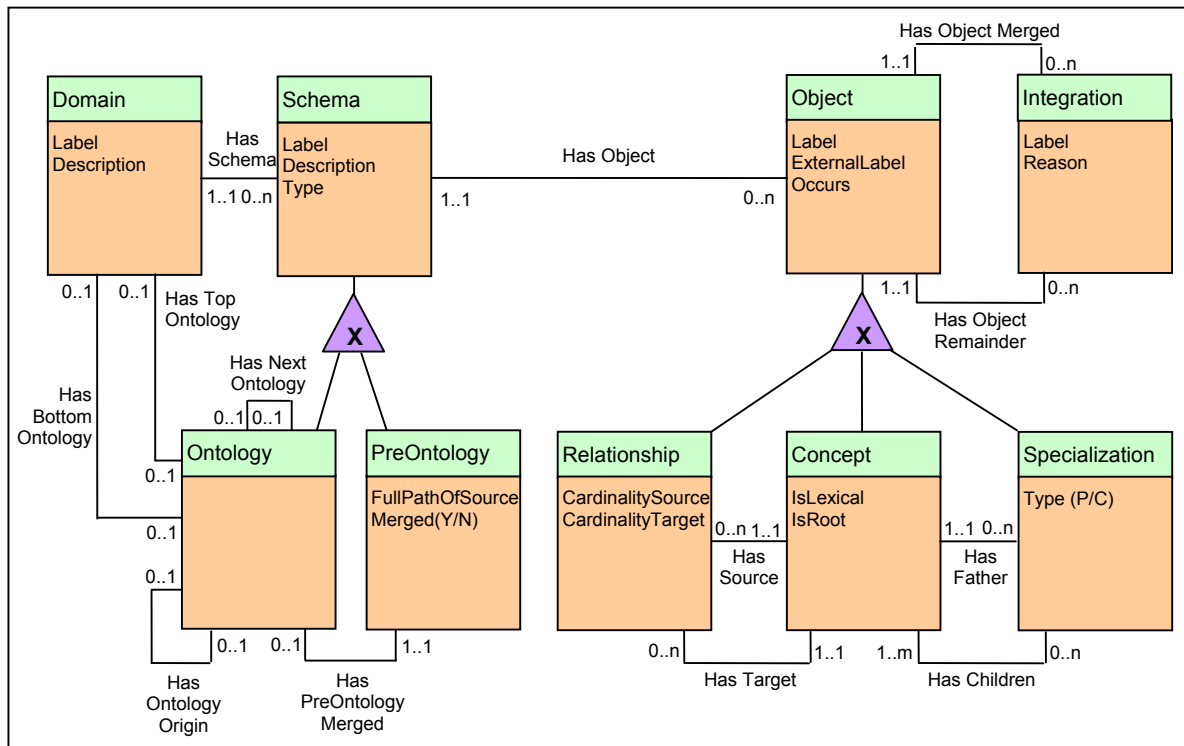


FIGURA 3.6 - Meta-modelo para pré-ontologias e ontologias.

3.3.1 Entidades

A figura 3.6, apresenta o modelo E-R elaborado para manter pré-ontologias e ontologias. As entidades e os atributos presentes no modelo são descritos nas próximas linhas.

1. *Domain*, registra todos os domínios de interesse e representa o ponto de entrada para todas as pré-ontologias e ontologias de cada domínio a serem mantidos pelo modelo, seus atributos são: *Label* que é o rótulo usado para identificar cada domínio e *Description* uma descrição textual sobre o mesmo.
2. *Schema*, contém informações sobre cada esquema (pré-ontologias e ontologias) que compõem um domínio. Seus atributos são: *Label* que é o rótulo utilizado para identificar cada esquema; *Description* que é um texto que descreve cada esquema e *Type* que identifica se o esquema é uma pré-ontologia ou uma ontologia.
3. *Ontology* (especialização de *Schema*) mantém as informações que são específicas de uma ontologia.
4. *PreOntology* (especialização de *Schema*) contém informações que são específicas de uma pré-ontologia, como: *FullPathOfSource* que contém o endereço completo da fonte de dados onde se encontra a DTD- XML (`//site.com.br/domíniox/artigo.dtd`) que dá origem a pré-ontologia e *Merged* que indica se a pré-ontologia já foi integrada a uma ontologia.
5. *Object* registra informações sobre os objetos que compõem um esquema (pré-ontologia ou ontologia). Seus atributos são *Label* que é o nome completo tal como se apresenta internamente na DTD origem, *ExternalLabel* é o rótulo a ser publicado para um objeto e *Occurs* que indica quantos objetos do mesmo tipo foram integrados até o então.
6. *Relationship* (especialização de *object*) contém toda informação necessária ao registro de cada um dos relacionamentos binários, componentes de um dado esquema. Seus atributos

- são: *CardinalitySource* que contém a cardinalidade de um dos conceitos participantes da relação e *CardinalityTarget* que contém a cardinalidade do outro conceito;
7. *Concept* (especialização de object) contém informações sobre cada um dos conceitos existentes em uma pré-ontologia seus atributos são *IsLexical* que indica se o conceito refere-se a um conceito léxico ou se se trata de um objeto complexo e *IsRoot* que indica o conceito raiz de uma pré-ontologia ou os conceitos raiz de pré-ontologias integradas em uma ontologia.
 8. *Specialization* (especialização de object) mantém informação sobre as especializações presentes em um dado esquema. Seu atributo único é *Type*, que indica se as entidades especializadas são mutuamente exclusivas ou não.
 9. *Integration* contém informações sobre a integração de dois objetos do mesmo tipo (conceitos ou relacionamentos), isto é, sempre que um objeto de uma pré-ontologia é integrado com um objeto de uma ontologia haverá uma entrada nesta entidade registrando a integração. Seus atributos são: *Label*, rótulo que identifica a entrada e *Reason* que identifica a regra do *thesaurus* que foi utilizada e que justifica a integração.

3.3.2 Relacionamentos

A figura 3.6, apresenta o modelo E-R elaborado para manter pré-ontologias e ontologias. Os relacionamentos entre as entidades do modelo e suas cardinalidades serão descritos a seguir.

1. *HasSchema: Domain has a Schema*, relaciona todos os esquemas ligados a um dado domínio. Um domínio pode não possuir ou possuir diversos esquemas associados a ele. Já um esquema está vinculado a um único domínio.
2. *HasSource: RelationShip has an Object Source*, indica um dos conceitos participantes de um relacionamento. Um relacionamento terá sempre um único conceito origem e um conceito pode não participar ou participar de vários relacionamentos.
3. *HasTarget: RelationShip has an Object Target*, indica o segundo conceito que participa de um relacionamento. Um relacionamento terá sempre um único conceito destino e um conceito pode não participar ou participar de vários relacionamentos.
4. *HasFather: Specialization has a Concept Father*, indica o conceito **pai**, ou seja, aquele mais abrangente e geral que está sendo especializado. Uma especialização deve possuir um único conceito **pai**. Já um conceito pode não participar, participar de uma ou de diversas especializações.
5. *HasChildren: Specialization has a Concept Child*, indica um ou mais conceitos **filhos**, ou seja, aqueles mais específicos e especializados com relação a um dado conceito **pai**. Uma especialização pode possuir um ou mais conceitos **filhos**. Já um conceito pode não participar, participar de uma ou de diversas especializações.
6. *HasObject: PreOntology has an Object*, relaciona todos os objetos pertencentes a uma mesma pré-ontologia. Uma pré-ontologia pode não possuir nenhum (uma pré-ontologia que esteja sendo recém introduzida) ou possuir diversos objetos. Já um objeto precisa necessariamente estar relacionado a uma única pré-ontologia.
7. *HasBottonOntology: Domain has a Botton Ontology*, indica a primeira ontologia do processo de integração. Um domínio pode não ter uma ontologia base ou ter apenas uma. Já uma ontologia pode não ser base de um domínio ou ser base de apenas um domínio.
8. *HasTopOntology: Domain has a Top Ontology*, indica a última ontologia do processo de integração, a de nível mais alto. Um domínio pode não ter uma ontologia base ou ter apenas uma. Já uma ontologia pode não ser a ontologia topo de um domínio ou sê-lo de apenas um.

9. *HasObject: Schema has an Object*, relaciona todos os objetos pertencentes a um mesmo esquema. Um esquema pode não possuir nenhum (um esquema recém introduzido) ou possuir diversos objetos. Já um objeto precisa necessariamente estar relacionado a um único esquema.
10. *HasOntologyOrigin: Ontology has an Ontology Origin*, relaciona uma ontologia com sua ontologia origem. Uma ontologia pode não possuir, no caso da ontologia base ou possuir apenas uma ontologia como origem. Já uma ontologia é origem da próxima exceto quando se tratar da última, no caso da ontologia topo.
11. *HasPreOntologyMerged: Ontology has a PreOntology Merged*, relaciona uma ontologia com a pré-ontologia com a qual foi integrada sua ontologia base. Uma ontologia sempre possui uma única pré-ontologia relacionada a ela e uma pré-ontologia está sempre relacionada a uma ontologia, exceto quando ainda não participou do processo de integração.
12. *HasNextOntology: Ontology has a Next Ontology*, relaciona uma ontologia com sua sucessora no processo de integração, indica a próxima ontologia, aquela da qual é base. Uma ontologia só não possui uma sucessora quando se tratar da ontologia topo.
13. *HasObjectRemainder: Integration has an Object Remainder*, indica o objeto remanescente da integração de dois objetos (um proveniente da ontologia base e outro proveniente da pré-ontologia que foi integrada). Sempre que houver a integração de um objeto de uma ontologia com um objeto de uma pré-ontologia o *object remainder* será o objeto da ontologia. Um elemento da entidade *integration* deve relacionar-se obrigatoriamente a um elemento da entidade *object*.
14. *HasObjectMerged: Integration has an Object Merged*, indica o objeto remanescente da integração de dois objetos e obrigatoriamente deve relacionar-se com um elemento da entidade *object*.

Uma vez descritas entidades, atributos e relacionamentos é importante ressaltar a existência de duas situações especiais no que se refere as especializações. A primeira é que o processo de integração não se dá sobre uma especialização. A verificação quanto a possibilidade/conveniência de inclusão de uma especialização é realizado sobre a ontologia e após o processo de integração, como pode ser observado pelo algoritmo na figura 7. A segunda situação diz respeito a origem. Como uma especialização não ocorre em uma pré-ontologia, mas sim sobre uma ontologia, uma especialização não possui uma origem como ocorre com conceitos e relacionamentos.

Feita a apresentação do meta-modelo e a descrição de seus componentes, a seguir, será apresentada a figura 3.7 contendo um exemplo de uma DTD para um e-mail (figura 3.7.a). A partir da DTD foi gerada sua representação gráfica⁶, onde pode-se perceber sua hierarquia de conceitos, bem como, o conceito raiz da árvore de conceitos. Na figura 3.7.c é apresentada a representação gráfica da pré-ontologia tal como será gerada no processo de mapeamento. Finalmente na figura 3.7.d é mostrada a pré-ontologia revisada e atualizada pelo usuário especialista, onde pode-se perceber que ele re-nomeou os relacionamentos (R1, ..., R5), muitos dos rótulos de conceito e algumas das cardinalidades. Pode-se perceber que a pré-ontologia mostrada na figura 3.7.d é muito mais rica semanticamente do que a mera representação gráfica da DTD, figura 3.7.b.

A seção 3.5.1 descreve o processo de mapeamento que a partir da DTD para um e-mail (figura 3.7.a) da origem a uma pré-ontologia (figura 3.7.c). Para ilustrar como ficam armazenadas as informações extraídas da DTD após o processo de mapeamento para pré-ontologia a figura 3.8, a seguir, mostra uma pré-ontologia instanciada.

⁶ Esta representação gráfica não é gerada ou manipulada pela proposta, e é apresentada com finalidade didática.

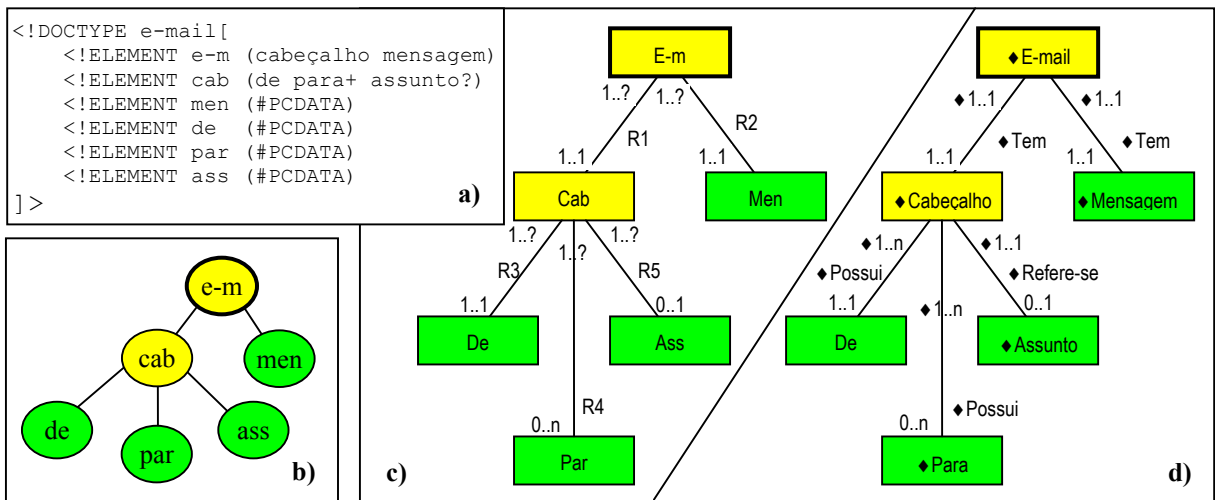


FIGURA 3.7 - DTD(a), representação gráfica(b), pré-ontologia(c) e pré-ontologia atualizada pelo especialista(d).

A figura 3.8 mostra o conteúdo das tabelas necessárias ao registro de uma pré-ontologia após ter sido realizado o mapeamento e após a intervenção do usuário especialista (figura 3.7.d). A DTD (figura 3.7.a) que foi mapeada para a pré-ontologia (figura 3.7.c) contém nomes de conceitos pouco significativos, não possui rótulos para os relacionamentos e não permite determinar as cardinalidades completas, estas incompletudes torna necessária a intervenção de um usuário especialista no domínio em questão para refinar a pré-ontologia antes que ela seja utilizada para compor a ontologia.

Label	Description	Type	FullPathOfSource
e-mail	Descreve os conceitos componentes de um e-mail	Pre	//inf.ufrgs.Br/documents/e-mail.dtd

a) Schema table

Label	ExternalLabel
e-m	◆e-mail
cab	◆cabeçalho
men	◆mensagem
de	de
par	◆para
ass	◆assunto
R1	◆Tem
R2	◆Tem
R3	◆Possui
R4	◆Possui
R5	◆Refere-se

b) Object table

RShip	CardinalitySource	CardinalityTarget
R1	◆1..1	1..1
R2	◆1..1	1..1
R3	◆1..n	1..1
R4	◆1..n	0..n
R5	◆1..1	0..1

c) Relationship table

Label	IsLexical
e-m	No
cab	No
men	Yes
de	Yes
par	Yes
ass	Yes

d) Concept table

Especialization	Type

e) Specialization table

FIGURA 3.8 - Instâncias para o meta-modelo da figura 3.6 referentes a pré-ontologia da figura 3.7.d.

Utilizando sua experiência e conhecimento do domínio este usuário utilizou rótulos mais significativos para os conceitos (e-mail, cabeçalho, mensagem, para e assunto) ao invés de e-m, cab, men, par, ass), substituiu os rótulos de relacionamentos (R1, R2, R3, R4 e R5 por Tem, Tem, Possui, Possui e Refere-se) e completou as cardinalidades que não haviam sido determinadas com precisão durante o mapeamento automático.

3.3.3 Thesaurus

Um *thesaurus*, no âmbito deste trabalho, é o objeto que descreve o que pode ser combinado automaticamente, bem como o que é permitido ser combinado manualmente. As entradas do *thesaurus*, figura 3.9, podem ser provenientes de três origens distintas.

- wordnet*⁷ [MIL 93 e BEC 93], inclui entradas de alto nível em idioma inglês que podem vir a permitir integrações de conceitos que envolvam este idioma. É necessário estabelecer um método de acesso como forma de garantir que entradas provenientes dele tenham o mesmo formato que as demais entradas.
- domain thesaurus* que abrange entradas específicas de um dado domínio e por isso deve ser consensual aos usuários com conhecimento daquele domínio.
- user's thesaurus* envolve entradas que são extensões de interesse a um usuário em particular. Estas entradas de um dado usuário, podem se tornar consensuais e por conseguinte passarem a integrar o *domain Thesaurus*.

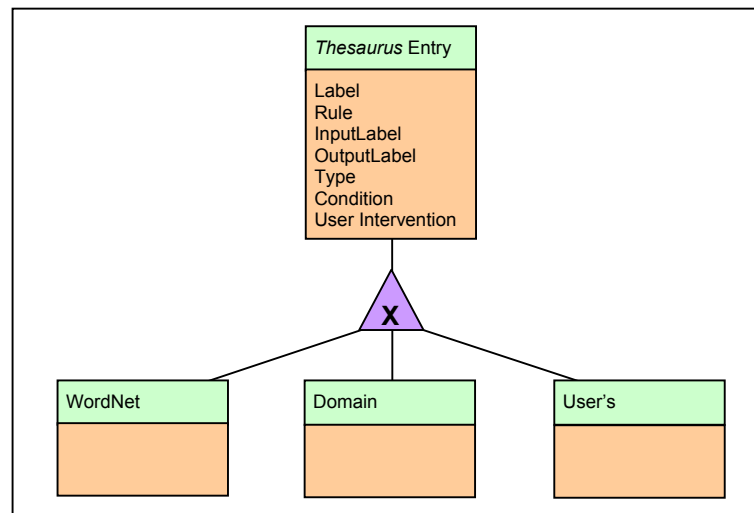


FIGURA 3.9 - O modelo de dados do *thesaurus* e suas possíveis entradas.

Uma possibilidade que pode ocorrer em virtude do uso de **dicionários** tão distintos quanto os citados é que um dado conceito pode ocorrer em mais de um deles e neste caso é necessário que seja estabelecido uma ordem de precedência entre eles. No caso deste trabalho, a ordem sugerida é que primeiro sejam usadas entradas do *user thesaurus*, depois do *domain thesaurus* e por fim do *wordnet thesaurus*. Um problema que pode ocorrer é de um dicionário conter uma entrada que se contraponha a outra entrada de outro dicionário. Neste caso o usuário precisa ser envolvido no sentido de resolver o conflito, uma vez que mesmo entradas conflitantes podem ser válidas em diferentes contextos. A indicação de uso de uma entrada conforme sua precedência, citada anteriormente neste parágrafo, pode equacionar este problema.

Neste trabalho a entidade *thesaurus* se compõe dos seguintes atributos:

- Label** é o identificador de cada entrada no *Thesaurus* e é utilizado pela entidade *Integration* para identificar a regra que justifica a integração de dois objetos;

⁷ O WordNet é um sistema de referência léxica disponível na *Web* cujo projeto foi inspirado por teorias psicolinguísticas da memória léxica humana. Palavras, verbos e adjetivos em inglês são organizados em conjuntos de sinônimos, cada um representando um conceito léxico. Sua biblioteca contém funções como FindTheInfo que com base em argumentos de entrada (palavra, categoria sintática e tipo de pesquisa) pesquisam sua base de dados retornando os resultado em um buffer ou estrutura de dados.

- b) **Rule** é um identificador de regras e uma regra pode ser composta por uma única entrada no *thesaurus* ou por diversas entradas associadas e identificadas pelo mesmo identificador, isto é, pelo mesmo conteúdo do atributo *rule*;
- c) **InputLabel** e
- d) **OutputLabel**, são os pontos de entrada no *Thesaurus*, que pode se dar nos dois sentidos, isto é, tanto pode-se obter o *output label* a partir do *input label* como o *input label* a partir do *output label*.
- e) **Type** identifica o tipo de relação existente entre o *Input* e *Output Label*, que podem ser: iguais, sinônimo, mais geral que (MGQ), mais específico que (MEQ) e regra de usuário. Rótulos que não possuam correspondência no *Thesaurus* serão considerados como não relacionados.
- f) **Condition** é um atributo fundamental quando a regra se refere a uma especialização e indica a condição que um conceito léxico deve verificar para que um conceito possa ser considerado a generalização de outro conceito especializado.
- g) **User Intervention** este atributo indica se a aplicação desta entrada do *Thesaurus* exige a anuência do usuário ou não.

A tabela 3.2, a seguir, apresenta alguns exemplos de entradas do *thesaurus* que foram utilizadas em alguns pontos deste trabalho. A duas primeiras entradas contém um mecanismo que dá suporte ao processo de análise/sugestão de estruturas de generalização/especialização. Este tipo de entrada permite utilizar um conceito léxico como determinante para que uma dada relação deste conceito com um outro conceito seja válido. Por exemplo, na primeira entrada do *thesaurus* abaixo (tabela 3.2) o rótulo de entrada **Sexo** será considerado sinônimo do rótulo de saída **Feminino** se a condição se verificar, isto é, se o conceito léxico **Sexo** contiver ou **F**, ou **Fem** ou **Feminino**. Na terceira, quarta e quinta entrada tem-se relações diretamente ligadas a estruturas de generalização/especialização, trata-se dos tipos mais geral que (MGQ) e mais específico que (MEQ), que são relações inversas. Portanto é preciso considerar que ao mesmo tempo que é dito que o conceito veículo é mais geral que o conceito moto deve-se ter em mente que, por consequência, o conceito moto é mais específico que o conceito veículo.

TABELA 3.2 - Exemplos de entradas do *thesaurus*.

Input Label	Output Label	Type	Condition	Rule	User Intervention
Sexo	Feminino	Sinônimo	Sexo = F, Fem, Feminino	1	✓
Sexo	Masculino	Sinônimo	Sexo = M, Masc, Masculino	1	✓
Veículo	Moto	MGQ			
Veículo	Carro	MGQ			
Ônibus	Veículo	MEQ			
Pessoa	Funcionário	Sinônimo			
Departamento	Setor	Sinônimo			

A estrutura definida para o *Thesaurus*, tabela 3.2, é bastante simples e foi projetada com a quantidade mínima de informação para dar suporte ao processo de mapeamento e integração. Desta forma espera-se que esta estrutura seja estendida em trabalhos futuros de forma a contemplar novas necessidades e possibilidades.

3.4 Mapeamento e Integração

Como citado inicialmente neste capítulo o processo de integração proposto se dá em duas fases. Inicialmente é realizado um processo de mapeamento das DTDs-XML para pré-ontologias que então serão integradas, uma a uma sobre uma ontologia topo. Esta seção visa apresentar e explicar estes dois algoritmos (mapeamento e integração), bem como mostrar como são resolvidos os conflitos de integração (nomenclatura), o processo de determinação de cardinalidades e de análise para uma possível introdução de estruturas de generalização/ especializações de conceitos.

3.4.1 O Algoritmo de Mapeamento

A figura 3.10 apresenta o algoritmo para mapeamento de esquemas XML (DTDs em específico) para pré-ontologias. Na figura a estrutura e o aninhamento de funções é mostrado explicitamente. O texto que segue explica em detalhe cada uma destas funções.

Uma vez definidos os esquemas XML que participarão do mapeamento o processo de integração é executado sobre cada um deles (figura 3.2 – 1ª. fase) e se apresenta da seguinte forma:

```

01. for each DTD in XML document source do
02.     Insert_Schema (new)
03.     for each root of tree or subtree do
04.         Rename_Concept (user's intervention)
05.         Insert_Concept
06.         for each subelement of tree or subtree do
07.             Rename_Concept (user's intervention)
08.             Insert_Concept
09.             Rename_Relationship (user's intervention)
10.             Insert_Relationship
11.             Review_Cardinality
12. Review_for_Concept_Specialization

```

FIGURA 3.10 - Algoritmo de mapeamento da pré-ontologia a partir de uma DTD-XML.

Para cada DTD-XML de interesse (linha 01) é **inserido um novo esquema** (linha 02) do tipo pré-ontologia. **Para cada conceito raiz de árvore ou sub-árvore** (linha 03) são realizados três grupos de atividades:

1. **Rename_Concept**, linha 04, onde o usuário tem a possibilidade de alterar o nome externo, ou seja, aquele que é publicado para o conceito;
2. **Insert_Concept**, linha 05, o conceito é inserido no esquema;
3. **Para cada sub-elemento da árvore ou sub-árvore** (linha 06), executam-se as seguintes cinco instruções:
 1. **Rename_Concept**, linha 07, o usuário tem a opção de alterar o nome externo do sub-elemento;
 2. **Insert_Concept**, linha 08, o conceito é introduzido no esquema;
 3. **Rename_Relationship**, linha 09, como os dois elementos componentes da relação já foram inseridos, o usuário tem a oportunidade de alterar o nome externo do relacionamento;
 4. **Insert_Relationship**, linha 10, em seguida o relacionamento será introduzido;

5. **Review_Cardinality**, linha 11, a cardinalidade da relação que envolve os dois conceitos será analisada e os atributos da *CardinalitySource* e *CardinalityTarget* da entidade *Relationship* serão completados.

Finalmente, em **Review_for_Concept_Especialization**, linha 12, será feita uma análise para verificar se é possível introduzir uma estrutura do tipo generalização/especialização na pré-ontologia. Este processo será descrito com mais detalhes na seção 2.5.4. mais adiante neste capítulo.

3.4.2 O Algoritmo de Integração

O algoritmo da figura 3.10, dará origem aos esquemas mapeados conforme apresentado na figura 3.2-1^a fase, estes mapeamentos serão utilizados, um a um no processo de integração, como mostrado na figura 3.2-2^a fase e cujo algoritmo é apresentado e descrito a seguir.

A segunda parte do processo é a integração propriamente dita (figura 3.11). Quatro grupos de instruções serão executados para cada pré-ontologia (linha 01) disponível para integração.

```

01. for each PreOntology Schema do
02.   Ontology (New) = Ontology (Top)

03.   for each Concept in PreOntology Schema do
04.     If Exist_Concept (Label, Ontology)
05.       Merge_Concept (Label, Ontology)
06.       Insert_Integration (RemainderLabel, MergedLabel)
07.     else If Concept_Synonym_Search (Label, Thesaurus, ListOfSyn)
08.       Merge_Concept_Synonym (ListOfSyn, Ontology)
09.       Insert_Integration (RemainderLabel, MergedLabel)
10.      else Insert_Concept (Label, Ontology)

11.   for each Relationship in PreOntology Schema do
12.     Review_Concepts in Relationship (Label, PreOntology)
13.     If Exist_Relationship (Label, Ontology)
14.       Merge_Relationship (Label, Ontology)
15.       Insert_Integration (RemainderLabel, MergedLabel)
16.     else If Relationship_Synonym_Search (Label, Thesaurus, ListOfSyn)
17.       Merge_Relationship_Synonym (ListOfSyn, Ontology)
18.       Insert_Integration (RemainderLabel, MergedLabel)
19.     else Insert_Relationship (Label, Ontology)

20.   Review for Concept Specialization

```

FIGURA 3.11 - Algoritmo de integração da pré-ontologia sobre a ontologia.

Primeiro será criada uma nova ontologia (linha 02), cujo conteúdo será originário da atual ontologia de nível topo, conforme indicado pela relação *HasTopSchema*. O segundo conjunto de instruções é executado para cada conceito existente na pré-ontologia (linha 03) que está sendo integrada e consiste basicamente em verificar a existência de cada conceito na ontologia. Se o conceito existe na ontologia (linha 04) ele será integrado (*Merge_Concept*, linha 05) e o registro da integração será inserido (*Insert_Integration*, linha 06). Se o conceito não existir na ontologia, será verificada a existência de um ou mais sinônimos para o conceito no *thesaurus* (linha 07). Havendo um ou mais sinônimos na lista será feita a integração deste

conceito sinônimo (*Merge_Concept_Synonym*, linha 08) e o registro desta integração será realizado (*Insert_Integration*, linha 09). Caso não haja sinônimos no *thesaurus* o conceito será inserido (*Insert_Concept*, linha 10) na ontologia.

O terceiro conjunto de instruções é executado para cada relacionamento existente na pré-ontologia (linha 11).

Inicialmente é feita uma revisão quanto aos dois conceitos pertencentes a cada relacionamento (*Review_Concepts*, linha 12). Este passo é necessário uma vez que quando da integração dos conceitos algum deles pode ter sido integrado e ter seu rótulo alterado, portanto o relacionamento existente e válido na pré-ontologia pode não ser mais válido após a integração. Esta revisão verifica compara os dois conceitos existentes em cada relacionamento com aqueles conceitos que foram integrados. Uma vez constatada a integração os conceitos originais serão substituídos pelos atuais conceitos remanescentes do processo de integração.

A seguir será descrito o processo de integração dos relacionamentos, que basicamente é uma adaptação ao processo de integração utilizado para conceitos. Se o relacionamento existe na ontologia (linha 13) ele será integrado (*Merge_Relationship*, linha 14) e o registro da integração será inserido (*Insert_Integration*, linha 15). Se o relacionamento não existir na ontologia, será verificada a existência de um ou mais sinônimos para o relacionamento no *thesaurus* (linha 16). Havendo um ou mais sinônimos na lista será feita a integração deste relacionamento sinônimo (*Merge_Relationship_Synonym*, linha 17) e o registro desta integração será realizado (*Insert_Integration*, linha 18). Caso não hajam sinônimos no *thesaurus* o relacionamento será inserido (*Insert_Relationship*, linha 19) na ontologia. Deve-se salientar que havendo relacionamentos sem nomes entre os mesmos conceitos nos esquemas a integrar (pré-ontologia e ontologia), estes serão integrados como se seus nomes (rótulos) fossem iguais.

Por fim e após terem sido analisadas as possibilidades de integração de conceitos e relacionamentos, será feita uma análise quanto a possibilidade de inserir na ontologia resultante alguma especialização (linha 20). Isto é feito analisando as regras pertinentes a especialização/generalização existentes no *thesaurus* frente aos conceitos e seus relacionamentos. Se houver correspondência a inclusão da estrutura de especialização/generalização pode ser realizada.

3.4.3 Conflitos de Nomenclatura

Durante o processo de integração procura-se determinar qual a relação existente entre cada conceito da pré-ontologia e cada conceito da ontologia sobre a qual se dará a integração. Esta proposta trabalha com cinco possíveis relacionamentos, ou seja, dois conceitos podem se relacionar por serem (a) iguais, (b) sinônimos, (c) um mais geral que o outro, (d) um mais específico que o outro e (e) não relacionados. Esta classificação é uma variação do proposto por diversos autores como Batini, Lenzerini e Navathe [BAT 86], Fang et al [FAN 91], Larson, Navathe, Elmasri [LAR 89] entre outros.

Sabe-se da literatura que o principal problema que afeta a integração de esquemas é o penoso processo de identificar e resolver conflitos [BAT 92]. Muitos dos conflitos apresentados na literatura não se aplicam na integração de esquemas XML, mas em outros casos a solução a ser adotada pode ser semelhante a utilizada na integração de esquemas de bancos de dados. O processo automático de integração se dá a partir da adoção de soluções tomadas emprestadas da literatura de integração de esquemas de BDs. A principal utilidade da

etapa automática de integração é desonerar o usuário de uma série de integrações conhecidas e triviais.

Conceitos que possuem o mesmo rótulo (nome) e relacionamentos que possuem o mesmo rótulo (nome) e ocorrem entre os mesmos conceitos podem ser integrados diretamente e oferecidos ao usuário para validação. Já conceitos ou relacionamentos com nomes distintos devem passar por uma etapa de verificação, onde se faz uma análise da relação existente entre eles e procede-se conforme o caso. Pares de objetos (conceitos e relacionamentos) cuja relação é:

- a) **igual** podem ser combinados diretamente;
- b) **sinônimo** são tratados como se fossem iguais, mas o rótulo resultante será o previamente existente na ontologia;
- c) **mais geral que** serão analisados sob a ótica da inserção de estruturas generalização/especialização conforme será discutido na seção 3.5.5;
- d) **mais específico que** serão analisados sob a ótica da inserção de estruturas generalização/especialização conforme será discutido na seção 3.5.5;
- e) **não se relacionam** serão integrados, isto é levados da pré-ontologia para a ontologia como são

3.4.4 Inferências e Conflitos de Cardinalidade

Definir a cardinalidade entre dois conceitos que se encontram relacionados através de aninhamento em uma DTD e que serão mapeados para uma pré-ontologia, envolve conhecer os símbolos que podem ser encontrados na DTD e que permitem determinar com relativa precisão a cardinalidade da relação do lado elemento filho do conceito. A cardinalidade do lado pai da relação pode, em alguns casos ser inferida mas deve ser validada pelo usuário. A revisão realizada sobre a cardinalidade da relação entre um elemento e um sub-elemento (linha 11 da figura 3.10) é realizada considerando os indicativos presentes na DTD conforme descrito a seguir:

- a) (+) indica que o sub-conceito ou grupo de sub-conceitos que antecedem o símbolo se repetirá uma ou mais vezes e será mapeada como uma cardinalidade 1..n;
- b) (*) indica que cada sub-elemento ou grupo de elementos que antecedem o símbolo ou não se repetirá ou se repetirá várias vezes e será mapeada como uma cardinalidade 0..n;
- c) (?) indica a opcionalidade do sub-elemento ou conjunto de sub-elementos e será mapeado como uma cardinalidade 0..1;
- d) a ausência de um símbolo que permita determinar a cardinalidade que indica que o sub-elemento deve aparecer uma única vez e será mapeado como 1..1.

O processo de determinar cardinalidades especificado acima permite deduzir a cardinalidade completa do lado sub-elemento da relação (elemento filho). O lado elemento (nodo pai) da relação, por não poder ser determinado com precisão será apresentada como 0..? quando entre o nodo pai e seus nodos irmãos ocorrer o indicador de opcionalidade (|) ou 1..? quando o símbolo (|) não ocorrer. A cardinalidade que é determinada automaticamente, conforme descrito acima, é posteriormente avaliada e confirmada pelo usuário.

Esta seção descreveu, até aqui, como utilizar símbolos presentes na DTD para inferir a cardinalidade de uma relação durante o mapeamento da DTD para a pré-ontologia. Um outro momento em que se faz necessário analisar a cardinalidade é durante a o processo de integração de uma pré-ontologia sobre uma ontologia. No processo de integração, quando

dois relacionamentos são integrados é necessário considerar as cardinalidades originais da pré-ontologia e da ontologia que estão sendo integradas. A cardinalidade resultante deverá ser sempre a mais ampla, isto é, a cardinalidade resultante (CR) da integração de uma cardinalidade **a** (CA) e uma cardinalidade **b** (Cb), será calculada da seguinte forma: $CR = \text{Min} (CMínA, CMínB) .. \text{Max} (CMáx, CMáxB)$ e que pode ser interpretada como segue: A cardinalidade mínima resultante (CMínR) será igual a menor entre as cardinalidades mínimas de A e B. A cardinalidade máxima resultante (CmáxR) será igual a maior entre as cardinalidades máximas de A e B. A figura 3.12, abaixo, demonstra a aplicação deste cálculo.

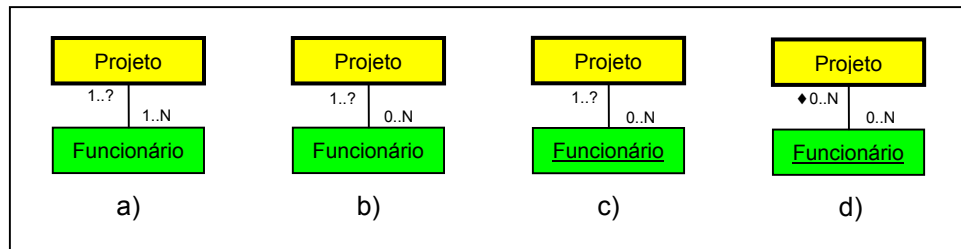


FIGURA 3.12 - Cardinalidades incompatíveis.

A figura 3.12 apresenta alguns exemplos que serão explicados a seguir: As figuras 3.12.a e 3.12.b ilustram o mapeamento de duas DTDs para pré-ontologias, onde a cardinalidade foi inferida com base nas seguintes linhas de duas DTDs distintas:

```
<!Element Projeto (Funcionário)+>
<!Element Projeto (Funcionário)*>
```

A figura 3.12.c apresenta o resultado da integração automática tal como gerado pelo algoritmo automaticamente. A cardinalidade resultante foi determinada considerando a menor entre as cardinalidades mínimas, ou seja **0** (zero) e a maior entre as máximas, ou seja, **N**. Já a figura 3.12.d apresenta o resultado gerado automaticamente revisado pelo usuário que completou a cardinalidade do elemento projeto passando-a de 1..? para 0..N.

3.4.5 Estruturas Generalização/Especialização

Neste trabalho o processo de determinar a possibilidade de inserir uma estrutura de generalização/especialização pode se dar em três momentos: (a) durante o mapeamento se o símbolo (|) ocorrer entre dois ou mais sub-conceitos este fato será tomado como um indicativo da possibilidade de introdução da estrutura, (b) no final do processo de mapeamento, comparando os conceitos presentes nas pré-ontologias com os presentes no *thesaurus* e (c) ao fim do processo de integração comparando os conceitos da ontologia com os presentes no *thesaurus*.

No primeiro caso, durante o mapeamento, a presença do símbolo (|) que indica a opcionalidade entre os sub-elementos de um dado elemento, pode indicar a possibilidade de introduzir uma estrutura de generalização/especialização. A figura 3.13 abaixo, mostra um trecho de uma DTD que contém o símbolo (|), o que pode indicar que os conceitos separados por ele são especializações do conceito pai e o resultado desta interpretação com a introdução da estrutura de generalização/especialização.

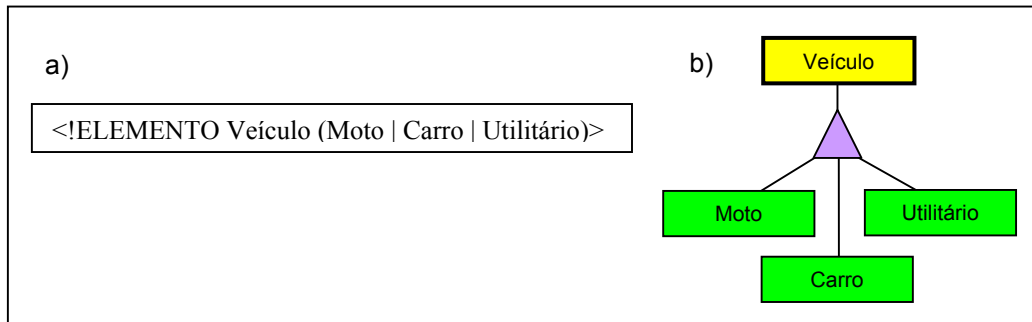


FIGURA 3.13 - Introdução de estruturas generalização/especialização.

A outra possibilidade de se introduzir uma estrutura de generalização/especialização pode ocorrer tanto para pré-ontologias quanto para ontologias quando a função *Review_for_Concept_Especialization*, linha 12 da figura 3.10 para pré-ontologias ou linha 20 da figura 3.11 para ontologias, for executada. Neste caso os conceitos presentes na ontologia ou pré-ontologia serão comparados entre si frente as relações entre conceitos presentes no *thesaurus*. Dois são os tipos de relacionamentos que interessam a esta função: (a) uma relação que diga que um conceito é mais geral que outro e (b) uma relação que diga que um conceito é mais específico que outro. Estas duas relações podem ser consideradas relações inversas entre si, isto é, se um conceito **X** for mais geral que um conceito **Y**, implica em que o conceito **Y** é mais específico que o conceito **X**. Se após a análise dos conceitos houver um ou mais pares de conceitos onde este tipo de relação ocorra, a relação direta entre eles pode ser substituída por uma estrutura de generalização/especialização.

Relação	Relação padronizada MGQ	Representação Gráfica
Pessoa MGQ Cliente	Pessoa MGQ Cliente	
Banco MEQ Pessoa	Pessoa MGQ Banco	
Fornecedor MEQ Pessoa	Pessoa MGQ Fornecedor	
Pessoa MGQ Funcionário	Pessoa MGQ Funcionário	
Transportador MEQ Pessoa	Pessoa MGQ Transportador	
Paciente MEQ Pessoa	Pessoa MGQ Paciente	
Pessoa MGQ Empregador	Pessoa MGQ Empregador	

FIGURA 3.14 - Análise de relações para introdução de estruturas generalização/especialização.

Assim se após a análise das relações entre conceitos de uma pré-ontologia ou de uma ontologia toma-se o conjunto dos pares de conceitos que tenham relações do tipo *mais_geral_que* (MGQ) ou *mais_específico_que* (MEQ), padroniza-se as relações utilizando a propriedade inversa, quando for o caso e submete-se o resultado a apreciação do usuário. A figura 3.14 ilustra o processo. Novamente, a participação do usuário, validando o processo automático é fundamental a qualidade do resultado final.

3.5 Tratamento de Conflitos – Estudo de Caso

O objetivo desta seção é apresentar alguns exemplos de resolução de conflitos que são citados em Batini, Lenzerini e Navathe [BAT 86] como conflitos semânticos típicos em se tratando de integração de esquemas de bancos de dados. Assim fragmentos de pré-ontologias serão integrados sobre fragmentos de ontologias, para mostrar o resultado da integração automática e o resultado das confirmações/atualizações realizadas pelo usuário no sentido de

agregar significado semântico ao modelo resultante. A intenção, com isto, é analisar como estes problemas clássicos são resolvidos na integração de esquemas proposta.

A figura 3.15, abaixo, mostra um conflito típico em integração de esquemas de bancos de dados convencionais. Trata-se da modelagem de uma mesma realidade sob diferentes pontos de vista o que normalmente ocorre quando a mesma realidade é modelada em locais distintos e por pessoas diferentes (em diferentes departamentos, países, etc.).

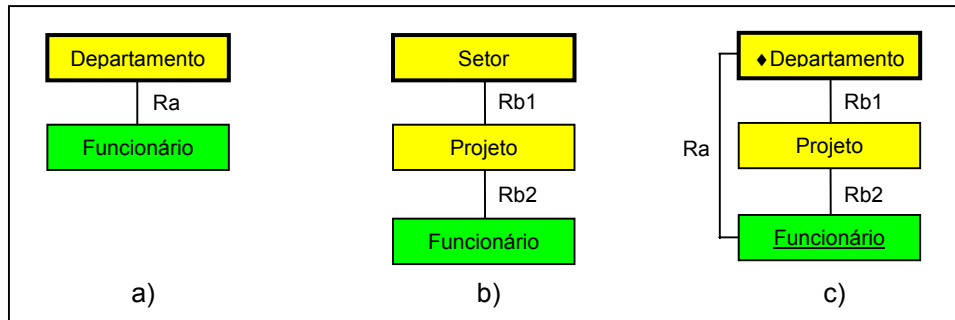


FIGURA 3.15 - Diferentes perspectivas (a e b) e o resultado da integração automática (c).

Na figura, diferentes fontes modelaram diferentemente a mesma realidade. Uma fonte relacionou os funcionários diretamente aos departamentos aos quais eles estão vinculados (figura 3.15.a). Uma outra fonte relacionou os funcionários aos projetos em que eles trabalham e relacionou estes projetos aos setores da empresa onde eles estão inseridos (figura 3.15.b). A figura 3.15.c mostra o resultado do processo de integração automática da pré-ontologia da figura 3.15.a sobre a ontologia da figura 3.15.b, considerando que o *thesaurus* possui uma entrada que coloca departamento como sinônimo de setor e que o usuário já revisou a integração realizada automaticamente. Conforme foi citado na seção 3.5.3, em caso de sinônimos o conceito resultante deveria ser o da ontologia, ou seja, setor e não departamento que é como ficou rotulado o conceito resultante. Deste exemplo pode-se inferir que o usuário interferiu no processo alterando o rótulo do conceito.

A seguir será apresentada uma visão parcial das principais tabelas utilizada no processo de mapeamento e integração. Também será feita a discussão das tabelas apresentadas procurando deixar claro o que foi feito em cada etapa. No capítulo 4 será apresentado um exemplo completo de mapeamento e integração, apresentando todas as tabelas envolvidas e o resultado após cada etapa do processo.

Para explicar o processo será utilizada a coluna passo. As primeiras duas linhas mostram o processo de mapeamento da DTD-XML para pré-ontologia, nesta etapa os conceitos e relacionamentos são mapeados tal como aparecem na DTD. Em primeiro lugar, passo 1a e 1b, os conceitos são introduzidos na tabela *concept* e em seguida, passo 1c, o relacionamento entre os dois conceitos é introduzido na tabela *relationship*. O mesmo processo ocorreu com a outra pré-ontologia e não é mostrado aqui. Deduz-se que a ontologia da figura 3.15 foi mapeada e em foi a primeira a participar do processo de integração, conseqüentemente ela é transferida para o modelo tal como se apresenta. A representação de seus conceitos podem ser vistas nos passos 2a, 2b e 2c e de seus relacionamentos nos passos 2d e 2e. Até este momento a tabela *integration*, tabela 3.5, esta vazia.

A próxima etapa consiste em integrar a pré-ontologia, figura 3.15.a sobre a ontologia, figura 3.15.b. As etapas necessárias serão descritas passo a passo a seguir.

TABELA 3.3 - A tabela Concept instanciada.

Passo	Schema	Label	ExternalLabel	Occurs	IsRoot	IsLexical	H.O.M.	H.O.R.
1a	PODep	LaDep	Departamento	1	Y	N		
1b	PODep	LaFun	Funcionário	1	N	Y		
2a	Oset	LbSet	Setor	1	Y	N		
2b	Oset	LbPro	Projeto	1	N	N		
2c	Oset	LbFun	Funcionário	1	N	Y		
3a/4i	OSet1	LbSet	Setor / ♦ Departamento	1/2	Y	N	LaDep	LbSet
3b	OSet1	LbPro	Projeto	1	N	N		
3c	OSet1	LbFun	Funcionário	1/2	N	Y	LbFun	LaFun

A integração de uma pré-ontologia sobre uma ontologia se inicia pela criação de uma nova ontologia contendo uma cópia da ontologia anterior. A seguir cada conceito da pré-ontologia será integrado sobre os conceitos presentes na ontologia. No caso em questão o conceito departamento (passo 1a) não existe entre os conceitos da ontologia, mas, como já foi citado anteriormente, o *thesaurus* possui uma entrada que coloca setor como sinônimo de departamento. Departamento é um conceito que existe na ontologia (passo 3a), portanto, a integração ocorre e o rótulo original (Setor) é mantido. Isto se dá por ser o comportamento padrão definido para o processo. O número de ocorrências do conceito é incrementado, os atributos H.O.M. e H.O.R. são definidos e uma entrada é inserida na entidade/tabela *integration* registrando a justificativa para integração.

A segunda e última integração de conceitos o conceito funcionário (passo 1b) existe na ontologia (passo 3c) e a relação entre estes conceitos indica que são iguais e portanto a integração pode ser realizada da mesma forma que ocorreu com o conceito anterior, apenas a justificativa na tabela *integration* será diferente.

TABELA 3.4 - A tabela Relationship instanciada.

Passo	Schema	Label	ExternalLabel	Occurs	Origin	Destiny	Card.S	Card.T	H.O.M.	H.O.R.
1c	PODep	Ra	Ra	1	LaDep	LaFun				
2d	Oset	Rb1	Rb1	1	LbSet	LbPro				
2e	Oset	Rb2	Rb2	1	LbPro	LbFun				
3d	Oset1	Rb1	Rb1	1	LbSet	LbPro				
3e	Oset1	Rb2	Rb2	1	LbPro	LbFun				
3h	Oset1	Ra	Ra	1	LbSet	LbFun				

Terminada a integração dos conceitos, passa-se a integração dos relacionamentos. A pré-ontologia em questão (figura 3.15.a) possui apenas um relacionamento (passo 1c - Ra) que por sua vez não existe na ontologia. Também não há no *thesaurus* nenhum sinônimo para Ra, conseqüentemente o relacionamento será incluído na ontologia.

TABELA 3.5 - A tabela Integration instanciada.

Passo	Label	Reason	Merged	Remainder
3f	I1	Sinônimos	LaDep (Departamento)	LbSet (Setor)
3g	I2	Iguais	LbFun (Funcionário)	LaFun (Funcionário)

As tabelas 3.3, 3.4 e 3.5, acima, mostram um instantâneo das principais tabelas do meta-modelo de integração contendo a situação apresentada na figura 3.15. As convenções de cores e atributos gráficos são os mesmos utilizados para ilustrar os modelos integrados.

Com relação a integração realizada deve-se ressaltar que nenhuma informação é perdida uma vez que a partir das informações mantidas pelo modelo integrado pode-se decompor uma eventual consulta em termos de conceitos e relações originais da fonte de dados. Assim, uma consulta que solicite, sobre o modelo integrado, os funcionários de um dado departamento pode ser decomposta em uma consulta que relacione os funcionários de

um dado departamento e em outra consulta que relacione os funcionários de qualquer projeto do departamento de interesse.

Outro problema comum segundo a literatura de integração de esquemas de BDs se dá quando uma mesma relação do mundo real é modelada utilizando diferentes construtores em diferentes fontes de dados. A solução para este caso no contexto de integração de esquemas de BD passa pela reestruturação dos esquemas o que acaba por ferir a desejável autonomia entre as fontes. No contexto de dados semi-estruturados, principalmente aqueles voltados para *Web*, retirar autonomia das fontes é uma alternativa fora de questão. De qualquer modo o esquema integrado que é desejado não requer o mesmo grau de rigidez que o necessário em integração de esquemas de BDs.

A figura 3.16, a seguir, mostra uma fonte de dados (a) que modela um conceito pessoa que tem como conceitos filhos os conceitos homem e mulher e estes tem um conceito filho chamado nome. Dependendo do conteúdo do *thesaurus* é possível que o processo automático sugira ao usuário inserir a especialização do conceito pessoa nos conceitos homem e mulher. A segunda fonte de dados (b) modela a mesma realidade com construtores distintos, ou seja, um conceito funcionário com dois conceitos filhos, no caso, nome e sexo. A figura 3.16.c apresenta o resultado da integração da pré-ontologia da figura 3.16.a com a ontologia da figura 3.16.b. O processo de integração automática combina os conceitos nome diretamente, uma vez que em ambos os esquemas os dois possuem o mesmo rótulo, já a integração do conceito pessoa sobre o conceito funcionário dá origem ao conceito funcionário no esquema integrado. Assim a única intervenção do usuário neste caso restringe-se a alteração do rótulo do conceito funcionário que recebe o rótulo pessoa.

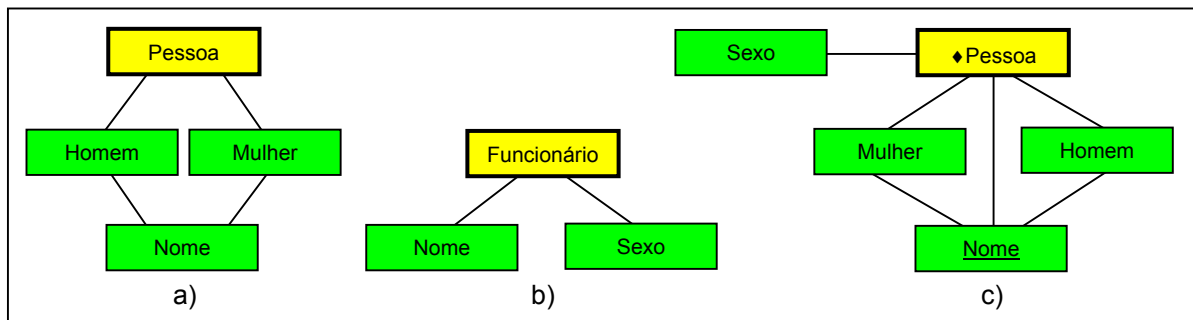


FIGURA 3.16 - Diferentes perspectivas (a e b) e o resultado da integração automática (c).

Da mesma forma que no exemplo anterior, neste caso também é possível decompor uma consulta realizada sobre o modelo integrado (figura 3.16.c) em termos de seus modelos componentes. Assim uma consulta que solicite os nomes de todas as mulheres terá de ser decomposta em uma consulta que solicite todos os nomes relacionados ao conceito **Mulher** no modelo da figura 3.16.a e em uma consulta que solicite o nome de todas as pessoas cujo conceito **Sexo** contenha **F** ou **Fem** ou **Feminino**. A consulta, muito provavelmente será construída sobre o modelo integrado (figura 3.16.c) e também é provável que haja uma indução em solicitar os nomes que tenham relacionamento com o conceito **Mulher**. Um mecanismo de decomposição de consulta em termos de conceitos fonte pode consultar o *thesaurus* (tabela 3.6, abaixo) a procura de do conceito **Mulher** utilizado na consulta original. Com base no *thesaurus* saberia-se que o conceito mulher é sinônimo do conceito sexo se este contiver **F**, **Fem** ou **Feminino**. Da mesma forma, se a consulta original solicitar o conteúdo do conceito nome para todas as pessoas que tenham o conceito sexo contendo **F**, **Fem** ou **Feminino** é possível utilizar o *thesaurus* para fornecer a informação de que quando o conceito sexo possui o conteúdo especificado na consulta e no atributo *condition* do *thesaurus* o conceito mulher pode ser tratado como sinônimo de conceito sexo. Aliando a isso o conhecimento mantido pelo modelo integrado de que o conceito sexo ocorre

em uma fonte e o conceito mulher em outra, consultas específicas podem ser construídas e os resultados combinados e informados ao usuário que os solicitou.

TABELA 3.6 - Instantâneo parcial do *thesaurus*.

Input Label	Output Label	Type	Condition	Rule	User Intervention
Sexo	Mulher	Synonym	Sexo = F, Fem, Feminino	1	✓
Sexo	Homem	Synonym	Sexo = M, Masc, Masculino	1	✓

O uso descrito acima também serve a decomposição de consultas que envolvam estruturas generalização/especialização no modelo integrado para chegar aos conceitos básicos nas fontes de dados.

O exemplo a seguir, baseado na figura 3.17 a seguir, apresenta uma situação onde ocorrem conflitos de nomenclatura. A pré-ontologia (figura 3.17.a) será integrada sobre a ontologia (figura 3.17.b). O conceito equipamento será integrado uma vez que a relação entre eles é de igualdade, já o conceito departamento possui um relacionamento com equipamento (proveniente de uma fonte de dados) que é distinto do relacionamento entre prédio e equipamento (existente em outra fonte). Contudo fazendo uma análise semântica dos conceitos departamento e prédio e de seus relacionamentos com o conceito equipamento, algumas conjecturas podem ser feitas e dentre estas algumas são citadas a seguir:

1. departamento e prédio podem ser integrados em um único elemento, tendo departamento ou prédio como rótulo, o que poderia ampliar o significado semântico do modelo.
2. setores de um determinado departamento podem estar espalhados por diversos prédios o que apontaria para uma ontologia como representada em 3.17.d.
3. em uma determinada estrutura administrativa, um prédio pode alojar todos os departamentos de uma organização fazendo com que a ontologia resultante se pareça com a apresentada na figura 3.17.e.

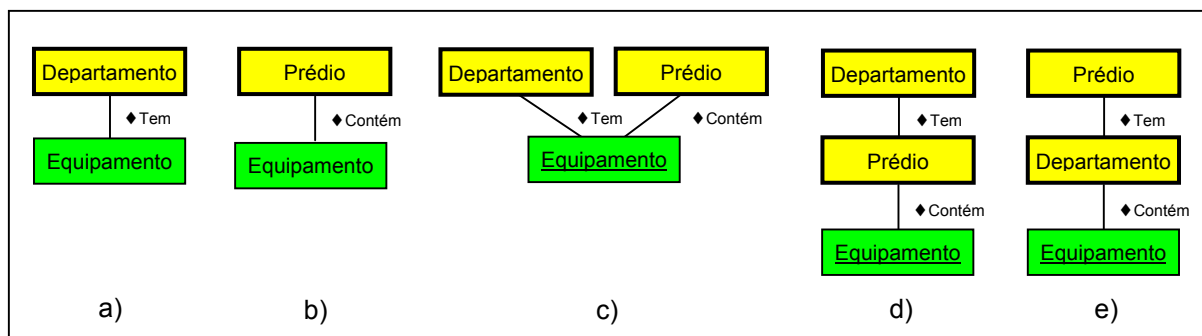


FIGURA 3.17 - Conflitos de nomenclatura(a e b), a integração automática(c) e outras possíveis representações (d e e) que possivelmente melhor representem semanticamente o domínio.

As possibilidades 1, 2 e 3 são apenas teorias, dentre várias outras possíveis, sobre uma realidade que somente pode ser confirmada por um usuário com conhecimento do domínio em questão. Este usuário pode acrescentar uma entrada ao *thesaurus*, que informe que departamento é sinônimo de prédio, como forma de estendê-lo para contemplar a automação da primeira possibilidade, já as possibilidades 2 e 3 não são possíveis uma vez que não existe um relacionamento entre departamento e prédio ou vice-versa e também porque estes dois conceitos encontram-se em fontes de dados distintas, não havendo relacionamento conhecido entre estes conceitos.

O próximo exemplo ilustra um conflito de nomenclatura bastante simples. Uma fonte possui um conceito chamado cliente relacionado a um conceito chamado crédito e outra fonte contém um conceito chamado consumidor relacionado a um conceito pedido. O conflito ocorre entre os conceitos cliente e consumidor, conforme a figura 3.18. Neste caso basta que o *thesaurus* contenha uma entrada indicando que cliente e consumidor são conceitos sinônimos para que a integração ocorra.

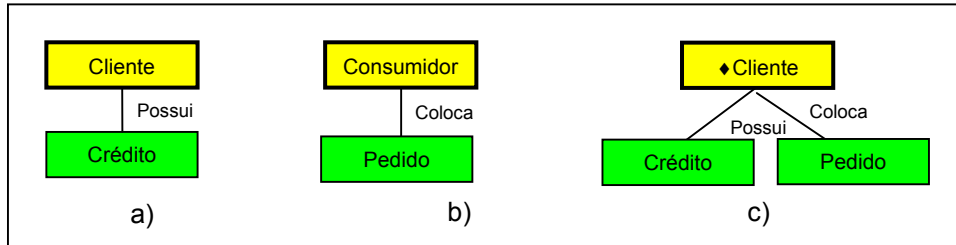


FIGURA 3.18 - Conflitos de nomenclatura (a e b) e o resultado da integração automática(c).

É importante observar, contudo, que o resultado da integração possui um único conceito comum as duas fontes e somente quando este conceito comum (cliente) for consultado é que o retorno poderá conter dados das duas fontes. Se uma consulta for realizada sobre os conceitos crédito ou pedido retornarão apenas conceitos relativos a fonte correspondente, isto é cliente ou consumidor (cliente após a integração) respectivamente. O resultado da integração automático manteria o conceito consumidor, conseqüentemente o usuário deve ter alterado o rótulo deste conceito visando a melhora semântica do modelo.

Estes exemplos procuram demonstrar o tratamento automático que ocorre durante o mapeamento e a integração, aproveita estes exemplos para ilustrar o tratamento dado a alguns conflitos clássicos presentes na literatura de integração de esquemas de BDs e, descreve também situações em que o processo de mapeamento/integração impede determinadas construções.

3.6 Intervenções do Usuário

O objetivo desta seção é apresentar os pontos onde o usuário especialista pode ou deve intervir. Não que isto ainda não tenha sido feito até aqui, em todas as oportunidades procurou-se sempre citar os pontos em que o usuário especialista no domínio em questão interage com a técnica apresentada e isto foi feito no momento em que o processo era apresentado, normalmente durante a descrição de algum ponto específico da técnica em que esta possibilidade existia. Nesta seção, entretanto, a intenção é explicitar estes pontos, o que ocorre a seguir.

O primeiro ponto onde o usuário deve participar do processo de integração se dá no momento em que ele seleciona as fontes de dados, isto é, a origem das DTDs que se deseja integrar. Uma vez definidas as fontes de dados o usuário deve definir entre os esquemas disponíveis (DTDs) aqueles que ele deseja mapear.

Após o processo de mapeamento o usuário tem a oportunidade de analisar e alterar os rótulos de conceitos, relacionamentos e cardinalidades das pré-ontologias. O usuário deve alterar rótulos que não expressem corretamente a semântica do conceito seja por que este rótulo esta abreviado ou porque não é o mais adequado. Os relacionamentos, em uma DTD, não são rotulados, assim é conveniente que o usuário os nomeie significativamente porque isto auxiliará, posteriormente, o processo de integração destes relacionamentos. Por fim as cardinalidades que são inferidas com base nas marcações contidas na DTD devem ser

revisadas e se for o caso atualizadas. É preciso considerar que apesar das cardinalidades de destino puderem ser apuradas com precisão as cardinalidades de origem não o são, assim é oportuno que antes de homologar o mapeamento de uma dada DTD ela seja tão refinada quanto possível.

Após o processo de integração o usuário tem a oportunidade de analisar e alterar rótulos de conceitos, relacionamentos e cardinalidades da ontologias resultante. É preciso que se considere que apesar da intervenção do usuário no processo de mapeamento ser fundamental a qualidade do resultado final, após o processo de integração é possível que rótulos de conceitos, rótulos de relacionamentos e cardinalidades em função do próprio processo de integração não tenham produzido o resultado que melhor traduza o significado desejado. Assim, é importante que após cada integração de uma pré-ontologia sobre uma ontologia o usuário revise o resultado e faça as alterações necessárias de forma que a ontologia resultante comunique o melhor resultado possível.

Conforme foi abordado na seção referente ao *thesaurus*, ele se compõem de entradas provenientes de diversas fontes, ou seja, entradas provenientes do WordNet, entradas provenientes do domínio em específico e entradas particulares daquele usuário. As entradas provenientes do WordNet são úteis a medida que os conceitos envolvam palavras no idioma inglês ou outro suportado pelo WordNet e o usuário deve determinar quais termos provenientes de lá incluir no *thesaurus*. Entradas no *thesaurus* em nível do domínio envolvem diversos tipos de termos. Se o processo de mapeamento ou de integração envolver palavras em outros idiomas entradas que permitam a tradução destas palavras para um idioma base serão necessárias e o usuário deve determinar quais participarão daquele domínio. Relações entre termos como sinônimo, mais geral que (MGQ) e mais específico que (MEQ) para um dado domínio são fundamentais e o usuário precisa inicialmente definir quais são e periodicamente revisá-las e completá-las. Por fim entradas que um dado usuário julga importantes para o processo de mapeamento e integração, mas para as quais não haja consenso quanto a conveniência de uso podem integrar um dicionário específico daquele usuário. Posteriormente estas entradas serão rotuladas como de domínio ou serão excluídas. O usuário também tem a responsabilidade de definir qual o tipo de relação existente entre cada par de conceitos, as condições que validam determinadas entradas, arrolar diversas entradas sob uma mesma regra e identificar quais entradas devem obrigatoriamente ser confirmadas pelo usuário.

O usuário deve durante o processo de integração identificar conflitos semânticos não resolvidos e utilizando as ferramentas que lhe foram fornecidas resolvê-los. Entres estes conflitos estão os sinônimos que é o uso de termos diferentes para identificar o mesmo conceito semântico, antônimos que é o uso de termos opostos e homógrafos que são termos com a mesma grafia mas que possuem significados distintos. Cabe ao usuário eliminar ou ao menos diminuir a incidência destes conflitos tão cedo quanto possível (preferivelmente na avaliação das pré-ontologias), uma vez que homógrafos, por exemplo, se não forem identificados a tempo podem ser combinados durante o processo automático, que integrará os dois conceitos dificultando que o problema seja percebido.

A última participação no processo confere ao usuário especialista o poder de decidir o nível de detalhe com o qual a ontologia resultante será publicada, isto é, se serão apresentadas especializações, rótulos de relacionamentos, cardinalidades e os identificadores de integração e de alteração realizadas pelo usuário.

O processo automático de integração tem como objetivo principal resolver boa parte do trabalho de tradução dos esquemas representativos de dados semi-estruturados em

pré-ontologias e integrá-las sobre uma ontologia e com isto desonerar o especialista no domínio de um processo repetitivo, cansativo e sujeito a erros. Contudo sabe-se que em diversas situações o mapeamento/integração realizado automaticamente não reflete o melhor resultado que pode ser obtido. Por isso é necessário fornecer ao especialista uma maneira de modificar as ontologias geradas automaticamente de forma a agregar qualidade ao modelo gerado.

3.7 Trabalhos Relacionados

A proposta discutida neste capítulo não surgiu do acaso. Como o ponto fundamental da proposta é gerar um modelo integrado para conjuntos de informação semi-estruturada, foi realizado um amplo levantamento bibliográfico da literatura de integração de esquemas de bancos de dados. Este levantamento foi realizado pelo simples motivo de o assunto estar sendo discutido a bastante tempo, desde meados dos anos 80, bem como, porque via de regra os temas referentes a bancos de dados são formalmente descritos. O interesse pela literatura de integração de esquemas de bancos de dados deu-se também, pela falta de literatura sólida e conceituada sobre integração de dados semi-estruturados cujo enfoque se desse na integração das estruturas que definem estes dados.

No levantamento bibliográfico realizado sobre dados semi-estruturados em geral e sobre dados XML, em específico, foram apontados alguns modelos para documentos XML, basicamente DTD e XMLSchema. Estes modelos servem como um esquema para documentos XML. Devido ao fato de estes esquemas para XML serem significativamente pequenos (quando comparados a esquemas de BDs tradicionais), e o domínio a que eles se referem ser restrito, espera-se ter um conjunto relativamente pequeno de esquemas XML a integrar. Como cada esquema XML refere-se a um dado domínio, que descreve os conceitos que os documentos vinculados aquele esquema abordam, espera-se que a integração destes esquemas forneça um esquema integrado representativo do domínio envolvido pelo conjunto de documentos representados.

Este trabalho considera o uso de ontologias na forma como descrito na integração de dados (seção 2.2) para propor que a integração de dados semi-estruturados seja feita com base na integração de pré-ontologias. Neste trabalho pré-ontologias são o mapeamento das DTDs XML (árvores de conceitos) para o modelo comum adotado (modelo de integração) que se baseia no modelo E-R (grafos de conceitos). Pré-ontologias representam os conceitos e relacionamentos entre conceitos existentes de uma única DTD mapeada. Portanto, este trabalho reserva ao termo ontologia um significado mais amplo, isto é, é o resultado da integração de um conjunto significativo de esquemas (pré-ontologias) o que faz com que a ontologia propicie um quadro amplo dos conceitos e relacionamentos entre conceitos existentes nas fontes de informação.

Com relação aos dados semi-estruturados, é fácil perceber que o assunto tem, há já algum tempo, ocupado um espaço significativo em face de sua adoção em larga escala por fornecedores dos mais diversos tipos de softwares (de linguagens de programação a bancos de dados) e pela comunidade científica. Assim, este trabalho abordou na sua seção 2.3, um resumo do tema dados semi-estruturados, em detalhe suficiente para permitir a compreensão dos conceitos, estruturas e ferramentas existentes e que são utilizadas na formulação desta proposta e principalmente no mapeamento da DTD para uma pré-ontologia.

A seguir citaremos diversos pontos pertinentes ao capítulo 2 deste trabalho, que referem-se a métodos de integração, características das ontologias e de dados semi-estruturados devidamente analisados sob a ótica deste trabalho. Esta análise é realizada, visto que, por exemplo, algo tido como problema em um método de integração de esquemas de BDs pode ser plenamente satisfatório na integração de esquemas de dados semi-estruturados.

- Em Bouguettaya, Benatallah e Elmagarmid [BOU 99] é colocado que uma desvantagem da integração global de esquemas (IGE) é que como não existe uma solução geral para os problemas de integração de esquemas (conflitos semânticos, estruturais e comportamentais), isto torna o processo dependente de especialistas humanos. Nesta proposta o especialista tem um papel fundamental, também, espera-se que os problemas semânticos sejam menores uma vez que o número de conceitos a integrar também o são.
- Um dos problemas associados ao processo de resolução de conflitos é que ele reduz drasticamente a desejada autonomia das fontes de dados (BDs relacionais) envolvidas [BOU 99], contudo como nesta proposta os esquemas originais não são afetados e a desejável autonomia é mantida.
- A decisão de integrar todos os esquemas a uma só vez ou dois a dois é uma decisão crítica [BOU 99]. Na primeira hipótese considera-se todo o conhecimento semântico disponibilizado pelos esquemas mas o processo torna-se mais complexo. Realizando a integração dos esquemas dois a dois o processo é facilitado pela diminuição do volume mas perde-se o todo semântico além de ter-se o problema de que a ordem em que os esquemas são combinados pode afetar o resultado final, o que não é desejável. O processo de integração aqui proposto realiza um processo incremental, onde cada pré-ontologia (esquema) é integrada sobre uma ontologia a cada vez. Espera-se que o tamanho reduzido dos esquemas e a participação ativa do usuário reduzam a indesejável possibilidade da ordem de integração afetar o resultado semântico final. Além disto, o usuário participa do processo de mapeamento da pré-ontologia e isto deve homogeneizar significativamente o resultado.
- Qualquer processo de integração de esquemas de BDs precisa considerar conflitos semânticos, conflitos estruturais, diferenças quanto ao modelo de dados usado suporte a sistemas, sistema operacional e diferenças quanto ao hardware [BAT 86]. Diferentemente da integração de esquemas de BDs a preocupação quando da integração de pré-ontologias reside principalmente nos conflitos semânticos e estruturais (estruturais em menor grau), uma vez que os demais conflitos não se aplicam porque os esquemas a integrar são homogêneos, isto é, estão expressos unificadamente como pré-ontologias.
- D. Fang et al [FAN 91] propõe uma arquitetura chamada *Remote-Exchange* que inclui um léxico local que é parte de um dicionário semântico. Já nesta proposta utiliza-se um *thesaurus* que se compõe de uma parte geral (tipo *WordNet*), uma parte específica do domínio e uma parte particular do usuário.
- Ao comparar conceitos Fang et al [FAN 91] utiliza os seguintes relacionamentos entre conceitos: idêntico, igual, compatível, tipo de, associação, coleção de, instância de, comum, característica e tem. Batini, Lenzerini e Navathe [BAT 86] utilizam idêntico, equivalente, compatível e incompatível. Gotthard et al [GOT 92] utilizam igual, subconjunto, sobreposição e disjunção como os possíveis relacionamentos entre dois objetos. Outros autores utilizam, ainda outros tipos de relacionamentos enquanto que nesta proposta utiliza-se os seguintes: igual, sinônimo, mais geral que, mais específico que e não relacionado.
- Ramash e Ram [RAM 95] sugerem usar propriedades dos objetos, como nomes de conceito, nomes de relacionamento e cardinalidades, para determinar os relacionamentos entre eles. Nesta proposta, são utilizadas basicamente estas mesmas propriedades para determinar conceitos e/ou relacionamentos relacionados e portanto passíveis de integração.

- Ram e Ramash [RAM 95], descrevem sua arquitetura de quadro negro, que se compõe de quatro níveis: dados, assertivas, fatos e metas. A proposta apresentada neste trabalho se assemelha a proposta por Ramash e Ram, uma vez que no nível de **dados** tem-se as DTDs que mapeadas tornam-se **assertivas** (pré-ontologias) que confirmadas pelo usuário tornam-se **fatos** que integrados geram uma assertiva (ontologia) que confirmadas pelo usuário, finalmente tornam-se **metas**.
- Shoval e Zohn [SHO 91] propõem uma metodologia que usa o modelo de relacionamento binário, faz uma integração binária e detecta automaticamente homonímias. A proposta apresentada neste trabalho mapeia DTDs diretamente para pré-ontologias, integra estas pré-ontologias uma a uma sobre uma ontologia, isto é, de forma binária e detecta automaticamente homonímias, tanto no nível de conceitos como de relacionamentos.
- Studer, Fensel, Decker e Benjamins [STU 99] diz que ontologias podem assumir 4 papéis distintos, e entre estes está a ontologia de domínio que captura o conhecimento válido para um tipo particular de domínio. Ontologias, na forma como são usadas nesta proposta, enquadram-se, segundo a classificação de Studer, como ontologias de domínio.
- Uschold e Gruninger [USC 96] apresenta quatro graus de formalismo para as ontologias, estes são altamente informal, semi-informal, semiformal e rigorosamente formal. O modelo conceitual utilizado é o E-R e o resultado gerado pode ser utilizado por um software, assim enquadra-se como rigorosamente formal.
- Segundo Uschold e Gruninger [USC 96] ontologias podem ser usadas para resolver problemas de comunicação, interoperabilidade e de engenharia de software. Nesta proposta ontologias são usadas para prover comunicação e interoperabilidade.
- Methontology [GOM 96], é um processo em etapas para construir ontologias de domínio. Esta proposta derivou do modelo de dados usado em Methontology o seu modelo de dados que se constituem de ontologias de domínio. Considerando o produto resultante de cada etapa de Methontology, esta proposta utiliza em maior ou menor grau os seguintes elementos: glossário de termos, árvore de classificação de conceitos, diagramas de relações binárias, dicionário de conceitos e tabela de relações binárias.
- A DTD permite identificar os elementos e sub-elementos que a compõe [SIL 02], bem como determinar (ao menos parcialmente) as cardinalidades entre eles. Para determinar as cardinalidades são usadas as seguintes notações da DTD (|, +, *, ?). Neste trabalho elementos de uma DTD-XML serão mapeados como conceitos complexos (não léxicos) e sub-elementos como conceitos léxicos. As cardinalidades parciais são determinadas durante o processo de mapeamento utilizando os símbolos citados acima.

3.8 Síntese do Capítulo

Este capítulo apresentou a proposta de mapeamento e integração propriamente dita. Do capítulo 2, a seção sobre ontologias serviu, entre outras coisas, de inspiração para o projeto do meta-modelo utilizado na proposta, da seção sobre integração de esquemas utilizou-se, basicamente, a taxonomia de conflitos e as relações entre objetos e da seção sobre dados semi-estruturados utilizou-se, não apenas, os símbolos que permitem determinar as cardinalidades dos relacionamentos. Já o capítulo 3, a proposta de mapeamento e integração, estruturou-se conforme descrito abaixo.

Para descrever a proposta de mapeamento e integração foi incluída uma seção que explica os objetos utilizados para representar cada um dos elementos componentes da metodologia, bem como o significado destes objetos. A seguir foi apresentado o meta-modelo projetado para manter pré-ontologias e ontologias, foi utilizado um modelo relacional com suporte a estruturas de generalização/especialização onde cada entidade, atributo e

relacionamento foi descrito. Ao fim da seção foi apresentado o *thesaurus* e a descrição de seus componentes, bem como, a descrição das origens que suas entradas podem ter.

As duas seções seguintes descrevem os algoritmos projetados para realizar o mapeamento (*wrapper*) de uma DTD-XML para uma pré-ontologia e para integrar uma pré-ontologia a ontologia. Estes algoritmos são apresentados e discutidos linha a linha. Os conflitos de nomenclatura e os tipos de relações entre os objetos são discutidos em uma seção a parte, apesar de logicamente fazerem parte do algoritmo de integração. Os conflitos de cardinalidade, também são discutidos em uma seção a parte, onde é mostrada a interpretação feita dos símbolos encontrados na DTD e da heurística utilizada para integrar cardinalidades conflitantes. Uma outra seção apresentando a forma utilizada para permitir que estruturas de generalização/especialização fossem introduzidas. A seguir são apresentados diversos exemplos de conflitos semânticos, obtidos junto a literatura que trata de integração de esquemas de BDs, bem como o tratamento dado a estes conflitos no contexto de integração de esquemas representativos de dados semi-estruturados. Os pontos em que o usuário é levado a interagir são explicitamente citados e descritos procurando deixar claro qual a participação do agente humano no processo. Finalmente o capítulo é encerrado pelo relato de outros trabalhos que influenciaram esta proposta, salientando pontos da literatura consultada que foram aproveitados.

4 Exemplo de Referência

O objetivo deste capítulo é o de apresentar um exemplo completo do processo de mapeamento e integração partindo das DTDs selecionada e progredindo passo a passo até obter-se a visão pública da ontologia.

O processo se inicia quando o usuário indica o caminho do local onde encontram-se as DTDs de interesse e seleciona dentre estas DTDs, aquelas que ele deseja que façam parte do mapeamento a ser realizado pelo *wrapper*. A figura 4.1, a seguir, apresenta as DTDs para *conference* e para *workshop* que serão utilizadas neste exemplo.

<pre><!-- Conference DTD Version 2.3 --> <!Element Conference (Name, Autor+)> <!Attlist Conference Year CDATA Place CDATA> <!Element Author (Name, eMail?, Paper, Affiliation*)> <!Element Name (#PCDATA)> <!Element eMail (#PCDATA)> <!Element Paper (Abstract, Body)> <!Element Abstract (#PCDATA)> <!Element Body (#PCDATA)> <!Element Affiliation (Institution Industry)> <!Element Institution (#PCDATA)> <!Element Industry (#PCDATA)></pre>	a)	<pre><!-- Workshop DTD Version 1.4 --> <!Element Workshop (Name, Writer+, email, BeginDate, EndDate)> <!Element Writer (eMail, Name, Institution, Tutorial)> <!Element Name (#PCDATA)> <!Element eMail (#PCDATA)> <!Element Institution (Address)> <!Element Address (#PCDATA)> <!Element Tutorial (Type, Duration)> <!Element Type (#PCDATA)> <!Element Duration (#PCDATA)></pre>	b)
--	-----------	--	-----------

FIGURA 4.1 - Representação textual das DTDs Conference (a) e Workshop (b).

As duas DTDs acima podem ser visualizadas graficamente conforme mostrado na figura 4.2, a seguir, que permite uma visualização dos conceitos, bem como, das relações entre eles. Esta representação é utilizada para apresentar as DTDs de uma forma gráfica. Esta representação não será utilizada nesta proposta, apenas foi incluída por ser mais intuitiva que a representação textual da DTD. O conceito raiz da DTD é representado através de uma elipse com as bordas mais espessas que os demais conceitos e a relação hierárquica deste com os demais é claramente percebida.

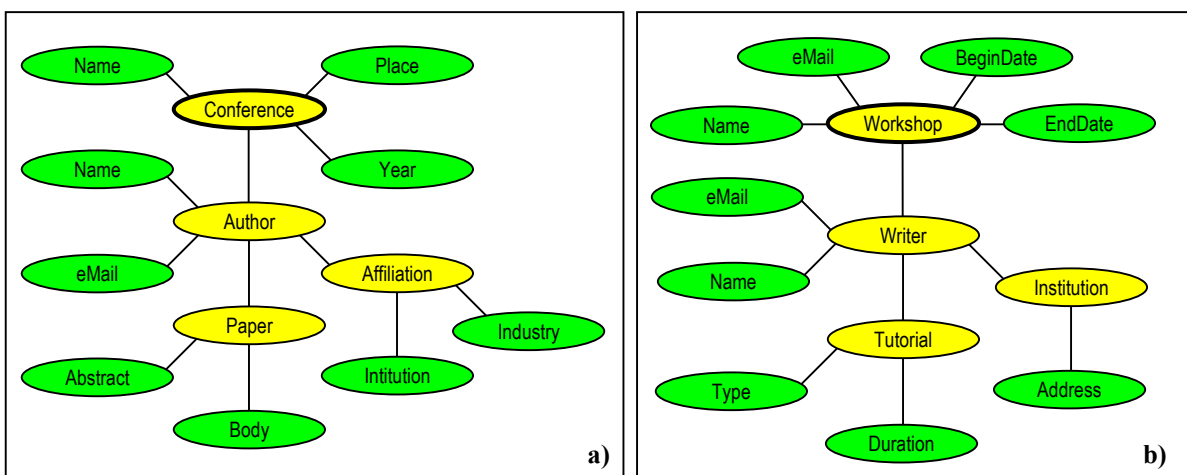


FIGURA 4.2 - Representação gráfica das DTDs Conference (a) e Workshop (b).

Uma vez que o usuário tenha definido as DTDs que participam do processo e dispare o processo de mapeamento, o *wrapper* passa a leitura de cada DTD navegando⁸

⁸ Esta navegação pode ser feita utilizando o método DOMParser como apresentado no anexo 1.

através dela e identificando cada conceito, relacionamento e inferindo (com base nos símbolos encontrados) as cardinalidades. Estas informações serão introduzidas no meta-modelo mostrado na figura 3.6. Após a conclusão do processo de mapeamento o usuário pode visualizar o resultado que será o mostrado na figura 4.3, abaixo. Deve-se notar que os conceitos e as relações entre eles são os mesmos que aparecem na DTD (figura 4.1) e mostrados na sua representação gráfica (figura 4.2). As mudanças automáticas significativas realizadas pelo processo de mapeamento residem no fato de que as DTDs passam a ser representadas (e armazenadas) através do modelo comum e que através da análise realizada sobre os símbolos presentes na DTD as cardinalidades foram inferidas. No que tange ao processo semi-automático o usuário tem a oportunidade de alterar os nomes (rótulos) dos conceitos ou introduzir nomes para os relacionamentos. Como não é prático proceder este tipo de alteração durante o processo de mapeamento conforme induz o algoritmo da figura 3.10, assume-se que ele seja realizado após o processo automático ter sido concluído.

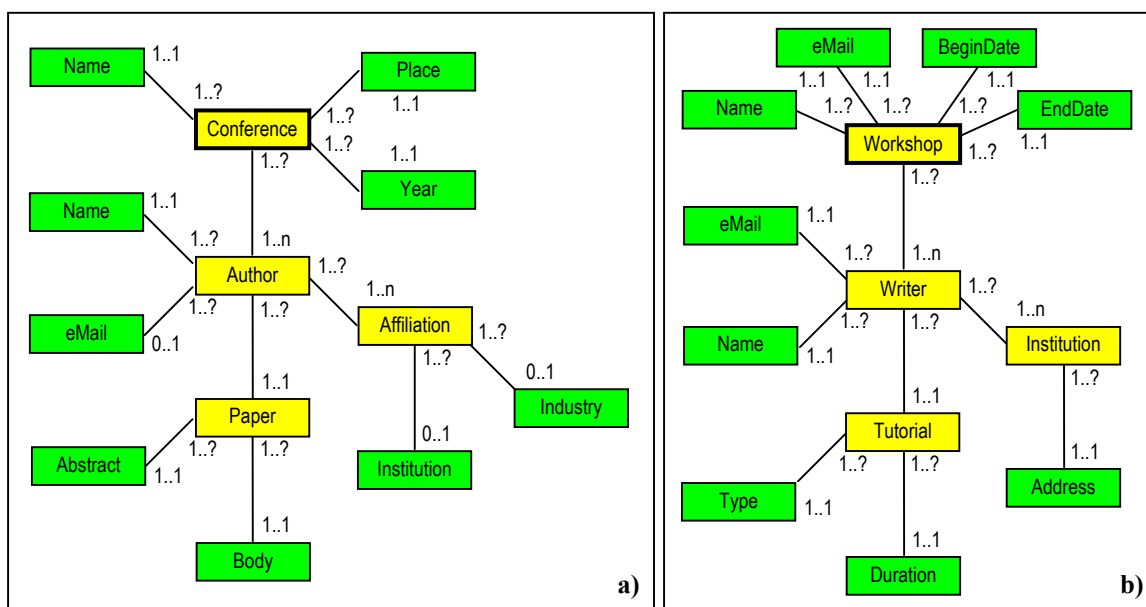


FIGURA 4.3 - Pré-ontologias mapeadas automaticamente das DTDs Conference (a) e Workshop (b).

A última etapa do processo de mapeamento é proceder automaticamente a análise quanto a possível introdução de estruturas generalização/especialização na pré-ontologia gerada. Na DTD *conference* (figura 4.3.a) e de acordo com a entrada 03 presente no *thesaurus* (tabela 4.1) *publication* é mais geral que *paper*, assim a generalização de *paper* poderia ter sido introduzida. Neste exemplo foi feita a opção por não introduzir a estrutura deixando esta alternativa para o processo de integração. Da mesma forma outro conceito mais geral poderia ter sido introduzido após o mapeamento da DTD *workshop* (figura 4.3.b), uma vez que o *thesaurus* contém a entrada 04 que diz que *publication* é um conceito mais geral que *tutorial*. Pelo mesmo motivo apontado anteriormente a decisão quanto a introdução da generalização foi deixada para o processo de integração.

TABELA 4.1 - Entradas do *thesaurus* necessárias ao mapeamento e integração.

Id	Input Label	Output Label	Type	Condition	Rule	User Intervention
01	Conference	Workshop	User's Synonym			
02	Author	Writer	Domain's Synonym			
03	Paper	Publication	Domain's MGQ		Ra	✓
04	Tutorial	Publication	Domain's MGQ		Ra	✓
05	Congregate	Contains	WordNet Synonym			

Esta análise também considera entradas do *thesaurus* classificadas como sendo do tipo sinônimo. Com isto o tipo mais geral que pode se dar não apenas entre os rótulos expressos literalmente, mas também entre sinônimos destes rótulos. Este uso combinado de conceitos do *thesaurus* amplia a capacidade automática do processo de inferir estruturas não incluídas originalmente nas DTDs.

A próxima etapa é proceder uma revisão, feita pelo usuário, sobre as pré-ontologias geradas automaticamente pelo processo de mapeamento. Nesta oportunidade o usuário pode alterar rótulos de conceitos, introduzir rótulos para relacionamentos e alterar as cardinalidades inferidas automaticamente durante o mapeamento. Neste exemplo e em acordo com a figura 4.4 o usuário procedeu as seguintes mudanças devidamente identificadas pela colocação de um losango a esquerda do objeto que sofreu alteração. Quanto a rótulos de conceitos apenas o conceito *affiliation* (figura 4.3.a) teve seu rótulo alterado para *entail*, como pode ser visto na figura 4.4.a; quanto aos rótulos de relacionamentos foram introduzidos, na figura 4.4.a, os rótulos *congregate*, *write* e *affiliated* e na figura 4.4.b os rótulos *contains*, *affiliated* e *presents*; finalmente quanto as cardinalidades diversas foram alteradas pelo usuário conforme indicado pelo losango a esquerda destas cardinalidades.

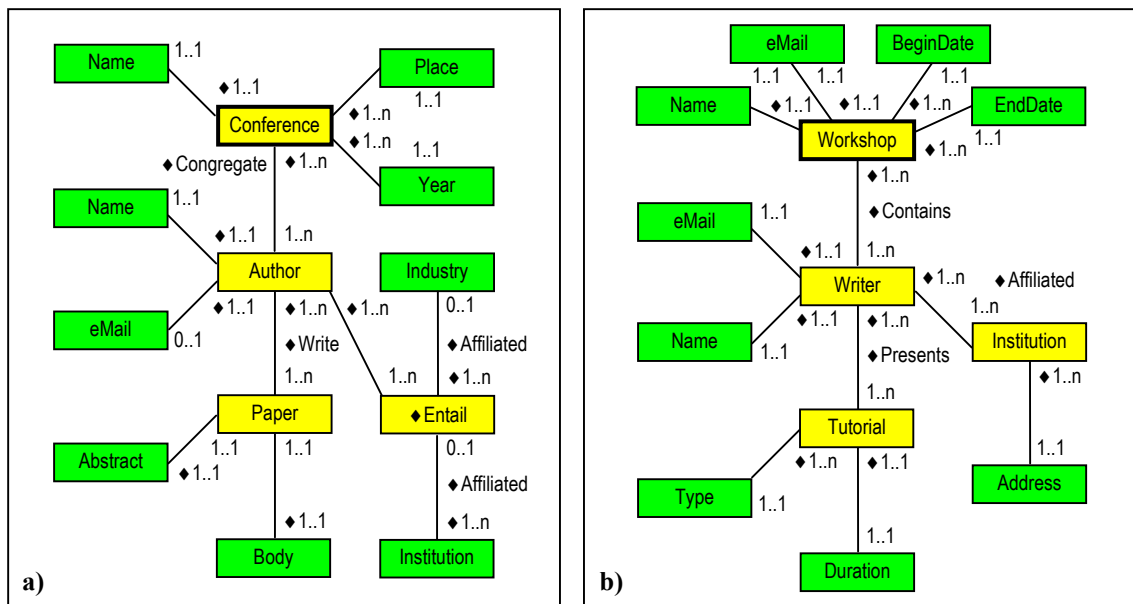


FIGURA 4.4 - Pré-ontologias Conference (a) e Workshop (b) atualizadas pelo usuário.

Após a revisão feita pelo usuário as pré-ontologias ficaram como mostrado na figura 4.4 e prontas para serem usadas como entrada para o processo de integração. As pré-ontologias serão integradas uma a uma, primeiro a pré-ontologia *conference* e em seguida a pré-ontologia *workshop*. Conforme indicado no algoritmo mostrado na figura 3.11 e como não existe, ainda, nenhuma ontologia gerada pelo processo de integração (indicado por *OntologyTop=null* ver figura 3.6), será criada uma ontologia vazia sobre a qual *conference* será integrada. Neste caso, em que uma pré-ontologia é integrada sobre uma ontologia vazia, pouca coisa é feita pelo processo de integração e a ontologia resultante do processo é muitas vezes exatamente igual a pré-ontologia que está sendo integrada. Assim a única integração que pode acontecer, ocorre se a pré-ontologia contiver conceitos com o mesmo rótulo ou com rótulos sinônimos que sejam filhos de diferentes conceitos, neste exemplo isto ocorre com os conceitos *conference* e *author* que tem como filhos o conceito *name*. Na figura 4.5, pode-se ver que tanto o conceito *conference* como o conceito *author* possuem um relacionamento com o conceito *name* resultante da integração. Sempre que uma integração é feita, uma entrada é incluída na tabela *integration*, como pode ser visto na entrada identificada como 01 da tabela

4.2. Desta forma tanto *conference* quanto *author* passam a ter relação com um único conceito *name* ao contrário do que ocorria com a pré-ontologia, conforme pode ser visto na figura 4.5.

TABELA 4.2 - Instantâneo da tabela *Integration*.

ID	Remainder	Merged	Reason
01	Name	Name	Synonym
02	Conference	Workshop	User's Synonym
03	Author	Writer	Domain's Synonym
04	Name	Name	Synonym
05	Email	Email	Synonym
06	Institution	Institution	Synonym
07	Congregate	Contains	WordNet Synonym

Também, na figura 4.5 abaixo, pode-se observar que o conceito *name* está sublinhado, esta é a identificação gráfica de que pelo menos uma integração ocorreu com aquele conceito.

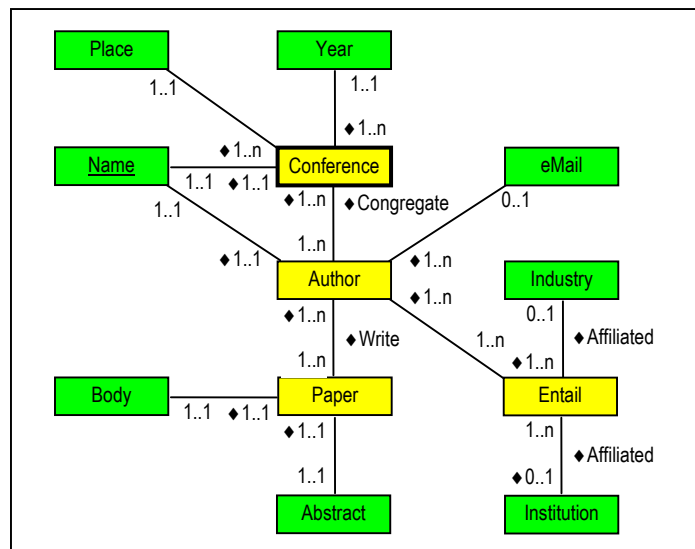


FIGURA 4.5-Ontologia resultante da integração da pré-ontologia *Conference*.

A integração da pré-ontologia *workshop*, bem como das próximas que porventura ocorram, será feita sobre uma ontologia existente. Assim pode-se esperar que conceitos e relacionamentos da pré-ontologia existam (direta ou indiretamente como no caso de sinônimos) na ontologia indicada por *OntologyTop*. O processo se inicia gerando uma cópia da ontologia topo sobre a qual será feita a integração da pré-ontologia. Em seguida cada conceito da pré-ontologia, iniciando-se pelo conceito raiz da pré-ontologia, será comparado com os existentes na ontologia de forma direta e se não houver coincidência direta será feita a comparação através dos sinônimos existentes no *thesaurus*. Deste processo e considerando o exemplo em questão pode-se identificar a ocorrência das seguintes situações:

- Conceitos que podem ser integrados diretamente: *name* e *email*;
- Conceitos que podem ser integrados indiretamente, ou seja, através da identificação de sinônimos com base no *thesaurus*: *Conference* e *Workshop*, *Author* e *Writer* e finalmente *Entail* e *Institution*;

- Relacionamentos que podem ser integrados diretamente: todos aqueles que não possuem rótulo ou que possuem o mesmo rótulo e que ocorrem entre dois conceitos existentes tanto na pré-ontologia quanto na ontologia. Considerando o exemplo em questão há o relacionamento *Affiliated* e;
- Relacionamentos que podem ser integrados indiretamente através do *thesaurus*: aqueles que como *Congregate* e *Contains* constam como sinônimos no *thesaurus*.

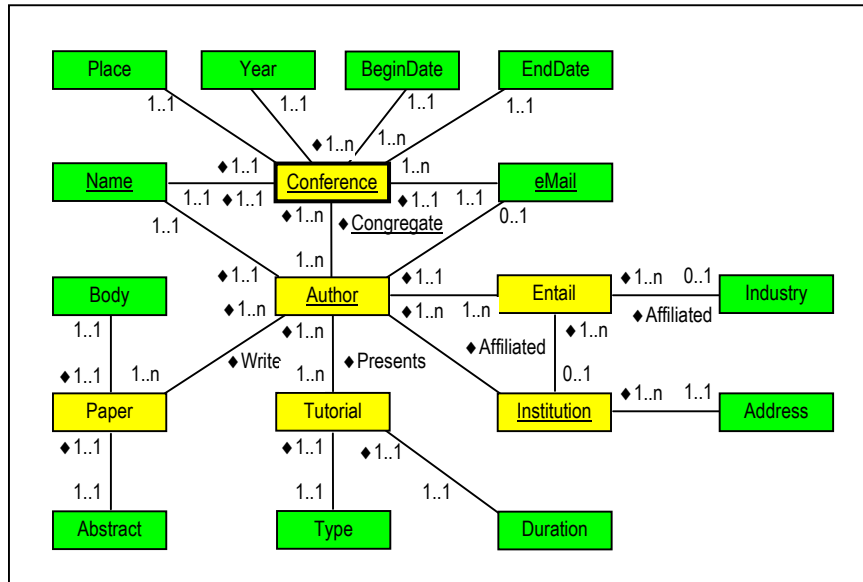


FIGURA 4.6 - Ontologia após sua integração com a pré-ontologia Workshop.

A figura 4.6, acima, mostra a ontologia resultante da integração automática das pré-ontologias *conference* e *workshop*. O passo seguinte, segundo o algoritmo, é proceder uma análise quanto a possibilidade de introduzir na ontologia uma estrutura de generalização/especialização. A figura 4.7, abaixo, apresenta a ontologia com a estrutura inserida.

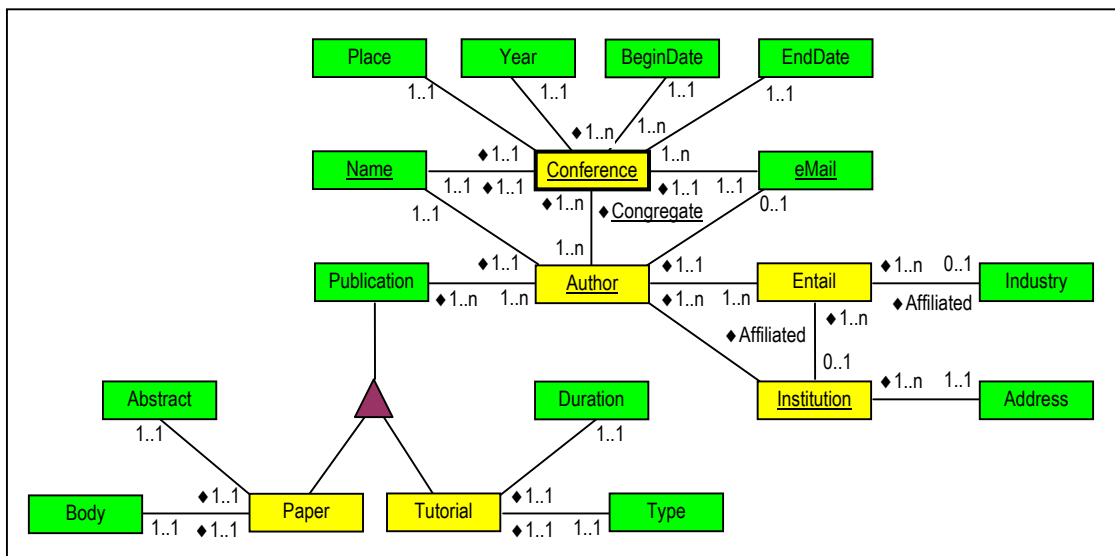


FIGURA 4.7 - Ontologia após a introdução da estrutura generalização/especialização.

A possibilidade de introduzir uma estrutura de generalização/especialização na ontologia apresentada na figura 4.6 é verificada com base em duas entradas do *thesaurus* (identificadas como 03 e 04), onde *Publication* aparece como sendo um conceito mais geral

que *Paper* e também mais geral que *Tutorial*. Isto torna possível a introdução da estrutura de generalização/especialização desde que o usuário permita esta ação, o que é exigido pelas duas referidas entradas do *thesaurus*.

A última representação prevista para o processo de mapeamento e integração é apresentada na figura 4.8, abaixo, que mostra o que se chamou de visão pública da ontologia. Esta representação é a mesma da figura 4.7 só que despojada dos nomes de relacionamentos e das cardinalidades destes relacionamentos. O objetivo desta representação é apresentar a ontologia com a quantidade de detalhe mínima mas suficiente a compreensão do domínio. Esta também parece ser uma forma de apresentação conveniente para, por exemplo, ser usada na formulação de consultas.

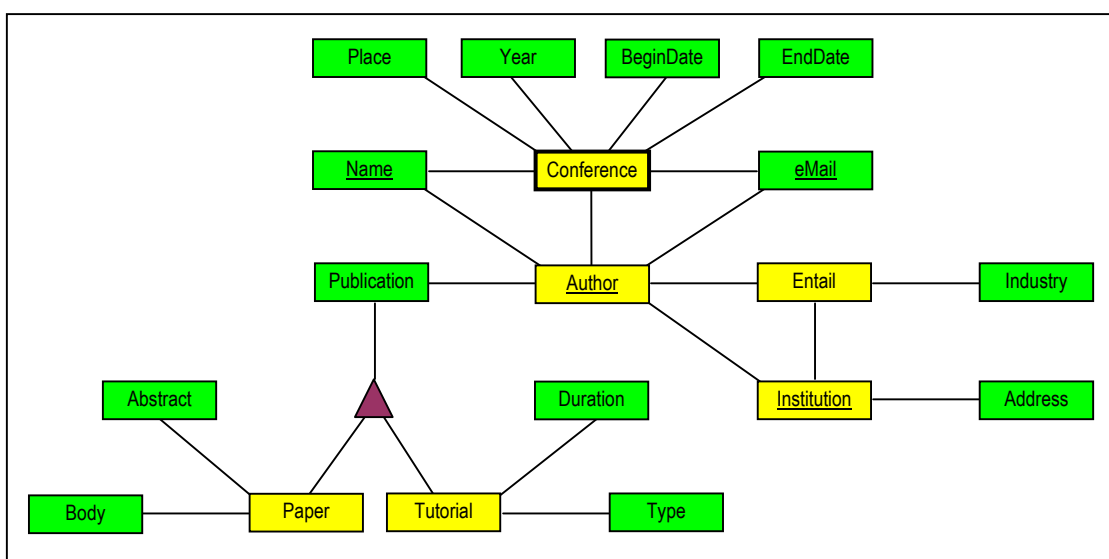


FIGURA 4.8 - Visão pública da ontologia resultante do processo de integração.

No que se refere a formulação de consultas, é importante salientar que isto não faz parte deste trabalho, mas sim dar condições a que isto seja feito. Neste sentido, a seguir, será dado um exemplo de como uma solicitação de consulta feita sobre a ontologia pode ser decomposta em termos de conceitos das pré-ontologias que lhe deram origem.

O exemplo de decomposição que será usado pede para que sejam relacionadas as **publicações de autores** com um dado **nome**. Utilizando SQL esta consulta se pareceria com o que é apresentado na figura 4.9.a. Esta solicitação foi feita considerando a visão pública da ontologia (figura 4.8) que é indicada por *OntologyTop* (ver figura 3.6) e sua decomposição poderia se dar da seguinte forma⁹.

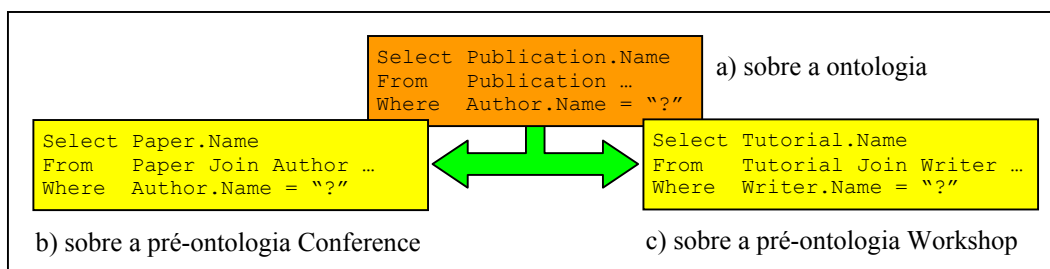


FIGURA 4.9 - Decomposição de consulta sobre a ontologia em termos das fontes.

⁹ A intenção deste exemplo é demonstrar que a decomposição pode ser feita e não como faze-la.

O primeiro passo consiste em identificar os conceitos envolvidos. No exemplo apresentado na figura 4.9.a são dois os conceitos de interesse: o nome da publicação (conceito *publication*) que é o resultado esperado para a consulta e o nome do autor (conceito *author*) que é utilizado como critério para seleção.

A forma para identificar os conceitos e suas fontes é percorrer o caminho que foi usado para gerar a ontologia de forma inversa, assim todas as etapas de integração devem ser consideradas. O processo só será encerrado quando não houver uma versão anterior da ontologia.

A ontologia que foi usada para compor a consulta, indicada por *HasOntologyTop*, tem sua origem na integração da versão anterior da ontologia, indicado por *HasOntologyOrigin* (ver a figura 3.6), com a pré-ontologia workshop, indicado por *HasPreOntologyMerged*. O conceito *publication* está presente na ontologia mas não na versão anterior da ontologia. Isto faz com que o *thesaurus* precise ser consultado. Nele encontra-se uma entrada onde o termo *publication* possui um relacionamento do tipo *mais_geral_que* com *paper* e outra entrada onde o termo *publication* possui um relacionamento do tipo *mais_geral_que* com *tutorial*. O termo *paper* é um conceito da versão anterior da ontologia e o termo *tutorial* é um conceito da pré-ontologia. No caso do conceito *tutorial* já é possível indicar parte da decomposição já no caso de *paper*, encontrado na versão anterior da ontologia, o processo precisa continuar. Neste ponto identifica-se que a pré-ontologia que foi usada na integração (*Conference*) contém o conceito *paper* e como não há uma versão anterior da ontologia (*HasOntologyOrigin = null*) o processo de decomposição para o conceito *publication* é encerrado.

Com o outro conceito envolvido, *author* é realizado o mesmo processo que indica que o conceito autor tem sua origem na pré-ontologia *Conference* e que *writer*, integrado a *author* por constarem no *thesaurus* como sinônimos, tem sua origem na pré-ontologia *workshop*. As alterações realizadas pelo usuário nos rótulos de conceitos não prejudicam o processo uma vez que o que é realmente alterado é o nome externo do conceito¹⁰, seu nome público, conseqüentemente toda informação necessária ao processo de decomposição é preservada.

Para finalizar este capítulo é conveniente citar que o mesmo processo descrito para decompor os conceitos em termos de seus formadores pode ser utilizado para decompor relacionamentos e as cardinalidades destes relacionamentos, uma vez que identificados, podem ser facilmente recuperados.

¹⁰ Ver os atributos da entidade *object* no meta-modelo presente na figura 3.6.

5 Conclusões e Trabalhos Futuros

A motivação para este trabalho reside na necessidade de mecanismos que facilitem o processo de gerenciamento de documentos que podem assumir os mais diversos formatos, mas que recentemente tem sido representados através das tecnologias de dados semi-estruturados, principalmente HTML e XML. O uso da *Web* em larga escala provocou uma grande profusão no volume de informação disponibilizado o que ampliou a necessidade deste tipo de mecanismo.

Uma abordagem baseia-se na realização de consultas sobre um modelo semântico que integra os conceitos de interesse e seus relacionamentos. Este modelo deve ser construído previamente por especialistas no domínio em questão (ver trabalhos relacionados em [MEL 01 e SAN 00]). Esta abordagem apresenta ao solicitante um modelo contendo as inter-relações entre os conceitos sobre o qual especifica-se tanto os conceitos que devem ser retornados, quanto os conceitos que devem ser utilizados como critério de busca. Esta integração precisa ser semântica, isto é, é preciso integrar conceitos que possuam a mesma intenção [GUA 98], o que propicia fornecer um volume menor de informação pré-processada e de melhor qualidade.

Dentro desta abordagem, diversos trabalhos de pesquisa têm proposto a criação prévia (por especialistas) de ontologias de domínio (que tem sido usadas como modelo conceitual para dados semi-estruturados [BEM 99]), bem como, a criação de um *middleware* que relacione os conceitos da ontologia as diversas fontes de dados de interesse [FEN 99 e GEN 97]. Este trabalho propõe que a criação da ontologia seja feita com base na seleção dos esquemas a integrar e que o mapeamento entre os conceitos da ontologia e os conceitos fonte seja feito automaticamente durante o processo de integração. O processo proposto gera, ao contrário de outros trabalhos, o histórico de todas as integrações realizadas. A ontologia gerada passa a atuar como um modelo conceitual para dados semi-estruturados assim como um modelo E-R age como modelo conceitual para BDs tradicionais. Dentre as fontes de dados semi-estruturados foi feita a opção pelo XML o que se deve ao nível de padronização alcançado, sua aceitação tanto nos meios acadêmicos como na indústria e pelo fato de que é estimulado que um documento XML esteja vinculado a algum tipo de esquema, tipicamente uma DTD ou XMLSchema [SIL 02 e W3C 02].

Os trabalhos de pesquisa abordados em Mello e Heuser [MEL 01] e Santi [SAN 00] constroem a ontologia previamente e então constroem uma camada de mapeamento entre os conceitos desta ontologia e os conceitos das diversas fontes. Ao invés disto, este trabalho propõe gerar a ontologia a partir da integração dos esquemas existentes, gerando o mapeamento de forma intrínseca ao processo. Este enfoque ascendente (*bottom-up*) de construção de uma ontologia de domínio tem três vantagens significativas, ou seja: (i) a ontologia terá apenas os conceitos efetivamente utilizados nos documentos de interesse tornando-se mais simples, (ii) o processo é alto-documentável sendo o mapeamento uma consequência automática do processo e (iii) o processo pode ser automatizado, ao menos parcialmente, o que reduz o esforço necessário para sua construção.

Com relação a bancos de dados, a considerou-se que os conflitos semânticos que ocorrem durante sua integração são semelhantes aos que ocorrem na integração de esquemas para dados semi-estruturados. Assim com base na literatura sobre integração de esquemas de bancos de dados [BAT 86, SHE 90, BOU 99, ELM 99, KAS 99, HAM 99, RAM 99 e FAN 91] chegou-se aos relacionamentos entre objetos utilizados nesta proposta, ou seja: (a) iguais, (b) sinônimos, (c) mais geral que, (d) mais específico que e (e) diferentes. Além dos

relacionamentos entre objetos foram relacionados diversos exemplos de conflitos o que foi apresentado na seção 3.5.

O objetivo geral desta proposta é gerar uma ontologia a partir da integração de esquemas representativos de dados que vão dos semi-estruturados aos estruturados. Esta ontologia passa a atuar como um modelo conceitual integrado dos diversos esquemas de interesse (DTDs a princípio) que servirá como ponto de partida para diversas atividades que podem ser realizadas sobre os dados representados por estes esquemas. Entre as atividades que esta ontologia deve prover suporte estão a extração e materialização de dados semi-estruturados, materialização de visões, processamento de consultas entre outras.

As principais características desta proposta de processo para a geração de uma ontologia de domínio são as seguintes.

- Define um processo de integração auto-documentável que permite: (1) tratar a evolução da ontologia em função de alterações e/ou inclusão de novos esquemas; (2) analisar a qualidade da ontologia gerada em função das entradas do *thesaurus*; (3) utilizar decisões anteriores usuário como forma de projetar decisões futuras e com isso ampliar o nível de automação do processo e; (4) gerar automaticamente o mapeamento entre conceitos.
- Define um meta-esquema para manter pré-ontologias e ontologias e define algoritmos de alto nível para o mapeamento da DTD para a pré-ontologia e para integração das pré-ontologias sobre a ontologia.
- Define as pré-ontologias como sendo o modelo comum de dados que permitirá o processo de tradução dos diversos tipos de esquemas mantendo a semântica original e permitindo a tradução do modelo comum para os esquemas originais conforme preconizado por Ram & Ramash [RAM 99]. Esta proposta utiliza um *wrapper* para mapear uma DTD para pré-ontologia, no futuro quando outros esquemas forem usados, XMLSchema por exemplo, novos *wrappers* deverão ser considerados.
- Define um modelo gráfico para pré-ontologias e ontologias baseado no modelo E-R sobre o qual foram incluídos identificadores de integração, de alteração pelo usuário, de identificação do conceito raiz e de identificação de conceitos complexos e léxicos.
- Define os pontos em que o usuário especialista pode intervir no processo, entre outros.

Dentre os pontos acima o primeiro constitui a principal contribuição deste trabalho. A proposta de um processo auto-documentável de integração, que considerando a literatura correlata não constitui um processo trivial, é o principal diferencial desta proposta frente as demais. Esta característica aliada ao uso do *thesaurus* ampliará a capacidade de processamento automático desonerando o usuário de uma atividade enfadonha, sujeita a erros e consumidora de tempo. A título de exemplo, Bouguettaya, Benatallah e Elmagarmid [BOU 99] dizem que mudanças nos esquemas locais normalmente implicam em que o processo de integração seja refeito total ou parcialmente para incorporar as alterações ocorridas. Como o processo é auto-documentável e o usuário administra as entradas do *thesaurus* quando o processo precisar ser refeito as decisões tomadas anteriormente podem ser reaproveitadas. O usuário pode então dedicar-se a seleção e avaliação dos esquemas fonte, a validação dos resultados gerados e a uma sintonia fina das entradas do *thesaurus*.

Ao mesmo tempo em que o estudo sobre integração de esquemas de bancos de dados tradicionais e sobre ontologias fornecia subsídios a integração das DTDs-XML, também era possível identificar pontos que poderiam ser melhor explorados, complementados, ou incluídos como extensões a esta proposta. Estes pontos são relacionados a seguir:

- A tese de doutorado de Mello [MEL 02] aborda em profundidade as possibilidades de mapeamento de DTDs. Nesta dissertação, o uso destas informações podem ampliar a capacidade de mapeamento automático das DTDs para pré-ontologias.
- A proposta feita neste trabalho aborda apenas as DTDs-XML como esquema a integrar, entretanto é conveniente que se incorporem outros esquemas para XML (XMLSchema, tipicamente), para HTML em função da quantidade de informação disponível neste formato, bem como, para dados estruturados.
- Esta proposta analisa apenas as relações diretas existentes entre conceitos e relacionamentos, contudo DeSousa [DES 86], Hayne e Ram [HAY 90] e Ramash e Ram [RAM 95] sugerem utilizar o grau de similaridade e dissimilaridade entre objetos para descrever os relacionamentos entre eles o que facilitaria a automação do processo, portanto, é conveniente que este enfoque seja abordado em trabalhos futuros de forma a estender esta proposta.

As extensões citadas acima permitirão aprimorar a atual proposta ampliando seu grau de automação, de representação semântica e a qualidade das inferências realizadas, contudo, novos trabalhos são publicados a cada dia e devem ser considerados ao estender o atual conteúdo desta proposta.

Anexo 1 Apresentação do Protótipo

O objetivo deste anexo é apresentar e descrever a implementação de um protótipo [MAR 00] que ocorreu no início deste trabalho de dissertação. Naquele momento havia muitos pontos indefinidos, definidos parcialmente ou ainda definidos de forma diferente da utilizada agora. Por exemplo, o termo pré-ontologia utilizado para identificar a DTD mapeada para o formato comum de integração, era, naquele momento rotulado por sub-ontologia que com o tempo mostrou-se inadequado. O uso do termo sub-ontologia pode ser observado em muitas das figuras apresentadas neste anexo e isto ocorre porque estas figuras refletem o protótipo que foi implementado naquele momento. Um outro caso de mudança no uso de termos pode ser observado em figuras onde aparece o termo ontologia local, atualmente utiliza-se apenas o termo ontologia, uma vez que o processo para gerar uma ontologia, uma ontologia local ou uma ontologia global é o mesmo, assim não se julgou conveniente utilizar um termo que restringe o escopo de abrangência da ontologia resultante.

A implementação que será descrita recebeu o nome de jOntology e sua implementação foi feita em Java. Onde era necessário interagir com os documentos XML foi utilizado um processador XML, o *Oracle XML Parser Versão 2* que possui um conjunto de classes Java para realizar o *parsing* de documentos XML e possui suporte ao modelo DOM (*Document Object Model*). A principal razão que levou a utilização deste *parser* em específico foi sua capacidade em suportar o *parsing* de DTDs o que era fundamental nesta implementação. A maioria dos processadores XML utilizam as DTDs apenas para validar o documento XML, não oferecendo recursos de acesso as DTDs. Já o processador escolhido possui um conjunto de classes para analisar sintaticamente uma DTD independente dos documentos XML vinculados a ela, com isso após o *parsing* a DTD fica representada como uma hierarquia de objetos o que facilitará o acesso. Além destes, foi também utilizado um pacote de classes denominado GFC4Java (*Graph Foundation Classes for Java*) da IBM/Alphaworks, que permite construir as representações dos grafos necessários a implementação. O protótipo descrito neste anexo possui os seguintes objetivos:

1. Fazer o mapeamento a partir das DTDs-XML;
2. Integrar os resultados dos mapeamentos de que trata o item 1;
3. Resolver problemas de integração como cardinalidades e sinônimos;
4. Permitir visualizar os resultados em um formato de grafo;
5. Permitir ao usuário editar os resultados;
6. Armazenar os resultados em arquivos XML de acordo com uma DTD.

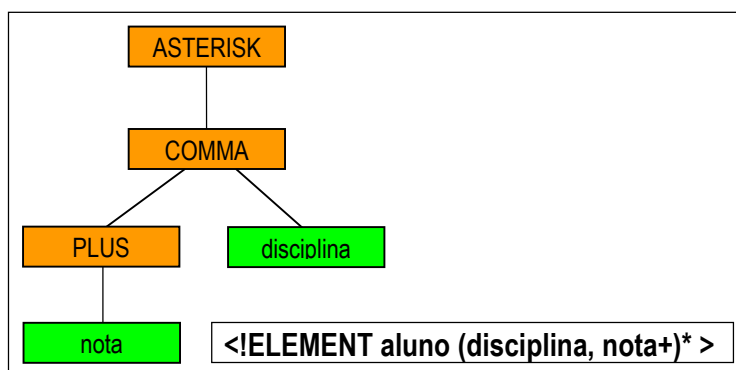


FIGURA A.1 - Utilização do método `getParseTree()` sobre uma DTD-XML.

Durante o processo de mapeamento o *parser* da Oracle que contém uma classe, DOMParser, cujo método parseDTD (URL) é utilizado para produzir o modelo de objetos da DTD especificada como parâmetro. Este modelo contém todos os objetos declarados como elemento da DTD. Um outro método, chamado getParseTree(), retorna uma árvore de nodos (objetos da classe Node). A figura A.1, mostrada anteriormente, demonstra isto.

O nodo chamado *comma* representa a vírgula, ou seja, uma seqüência de declarações de elementos para o elemento aluno. Os nodos *asterisk* e *plus* representam os sinais de ocorrências dos elementos declarados e por fim *nota* e *disciplina* representam os elementos declarados para o elemento **aluno**. A partir desta árvore é que será possível determinar os conceitos, os relacionamentos entre conceitos e as cardinalidades destes relacionamentos.

A figura A.2, a seguir, será utilizada para descrever a interface do protótipo bem como do mapeamento de duas DTD para sub-ontologias e da integração destas duas sub-ontologias em uma ontologia local.

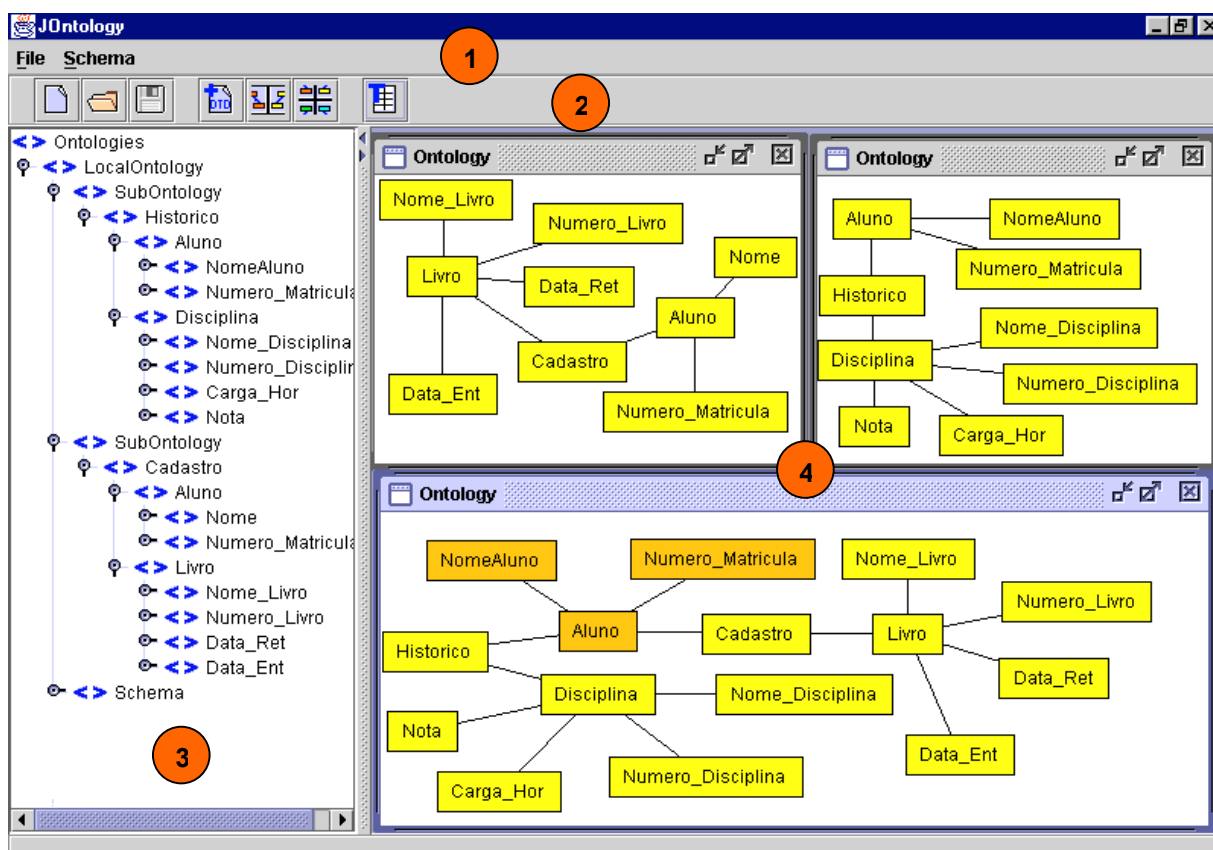


FIGURA A.2 - Sub-ontologias cadastro e histórico e a ontologia local resultante.

A interface do protótipo jOntology foi projetada para ser tão familiar, simples e funcional quanto possível e como requer um protótipo. Para implementá-la foi utilizado o pacote de classes Swing que é parte da implementação Java utilizada.

A área horizontal onde se situa o círculo identificador 1, como em qualquer aplicativo que use o padrão de janelas destina-se ao menu principal de opções. O menu principal do protótipo contém duas opções, a opção *File* e a opção *Schema*. A opção *File* contém as sub-opções *New Ontology*, *Open Ontology* e *Save Ontology as XML*. Já a opção *Schema* possui as sub-opções *Add DTD*, *Integrate Selected* e *Integrate All*.

A área horizontal onde foi colocado o círculo identificador 2, também como muitos aplicativos destina-se a botões de acesso rápido as principais funções do protótipo. As funções dos botões de acesso rápido da esquerda para a direita são: criar uma nova ontologia, abrir uma ontologia existente, salvar a ontologia atual (em formato XML), criar uma sub-ontologia (a partir de uma DTD), integrar os esquemas selecionados, integrar todos os esquemas e visualizar e/ou editar o *thesaurus*.

O círculo identificador 3, indica uma parte da área de trabalho construída utilizando um objeto Swing do tipo *jTree*, que se destina a criar representações hierárquicas no tradicional formato de pastas (como utilizados em diversos gerenciadores de arquivos). Nesta área o usuário pode selecionar, editar ou eliminar um objeto.

Finalmente o círculo 4, identifica a área onde os esquemas são apresentados. Esta área foi construída utilizando, também, um objeto Swing chamado *jDesktop*. Esta área pode conter diversas janelas internas que serão utilizadas para apresentar as representações visuais das sub-ontologias e das ontologias

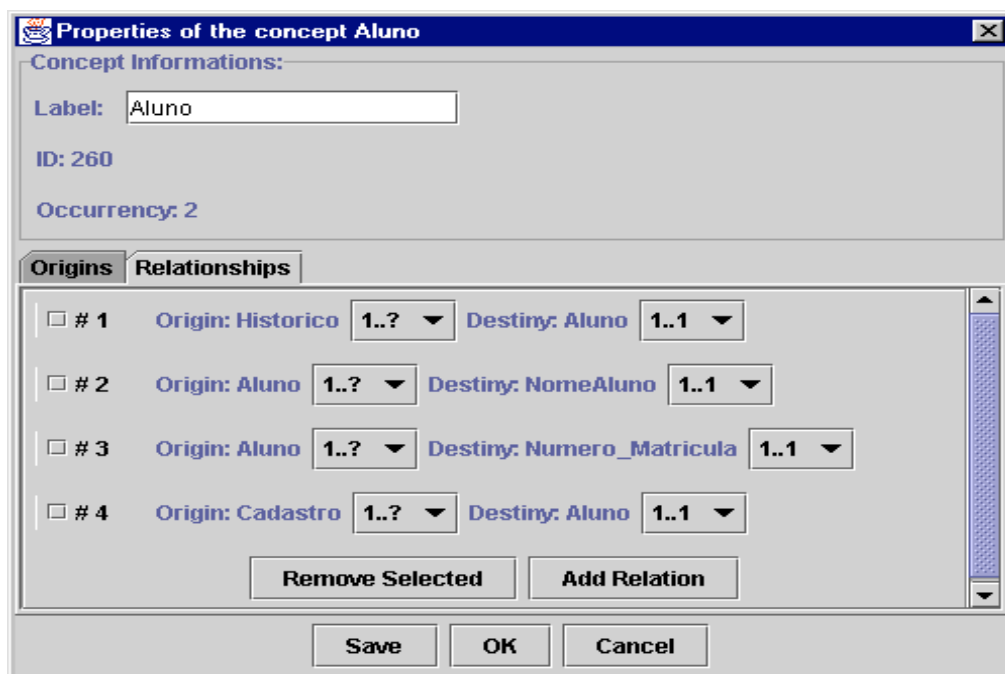


FIGURA A.3 - Propriedades do conceito aluno.

As propriedades de um conceito podem ser consultadas ou editadas através da árvore de conceitos ou através de sua representação gráfica. Esta operação realizada sobre o conceito **aluno** apresenta a janela mostrada na figura A.3. Nesta janela o usuário pode editar o rótulo do conceito, seus relacionamentos, as cardinalidades destes relacionamentos e as suas origens, fontes de dados onde o conceito ocorre.

A figura A.4, mostra o *thesaurus* rudimentar que foi utilizado para a criação do protótipo. Este *thesaurus* continha apenas sinônimos que eram acessados através de um rótulo de entrada e que devolvia um rótulo de saída ou ainda uma lista de rótulos de saída o que ocorreria caso houvesse mais de um rótulo de entrada para o argumento de pesquisa fornecido.

Para encerrar este anexo será feita uma descrição sucinta das etapas necessárias para executar algumas funções como criar uma nova ontologia, integrar esquemas criando uma ontologia local e salvar no formato XML, utilizando o protótipo jOntology.

Para criar uma nova sub-ontologia o usuário irá pressionar o botão correspondente (o quarto botão da esquerda para direita) ou selecionará *file* no menu de opções e em seguida selecionará *new ontology* no sub-menu. O programa irá requisitar o caminho da DTD que deverá ser mapeada para uma sub-ontologia. Após validar o caminho, o programa realiza o mapeamento e inclui a nova sub-ontologia no jTree e no jDesktop.

Input Label	Output Label
autor	author
Nome	NomeAluno
NomeAluno	Nome
B	D
D	B
B	H
H	B
G	J
J	G
C	I
I	C

FIGURA A.4 - *Thesaurus*.

Para integrar esquemas e criar ontologias locais o usuário tem duas opções: selecionar (com um right ou shift click sobre o nodo) os esquemas de interesse e então integrar os selecionados (utilizando o quinto botão) ou integrar todos os esquemas (sexto botão). Qualquer uma das alternativas irá gerar uma nova ontologia local onde os conceitos que foram integrados serão representados na cor laranja, enquanto que os que não foram integrados são representados em amarelo.

Para salvar arquivos em formato XML o usuário deve utilizar o terceiro botão ou a opção file seguida da opção *save ontology to XML*. O protótipo solicitará o caminho e o nome do arquivo a gerar, e utilizará o método `print(URL)`, que é definido na classe `Document` do DOM, para gerar o arquivo conforme o estado atual da ontologia

Bibliografia

- [ABI 97] ABITEBOUL, Serge. Querying Semistructured Data. In: INTERNATIONAL CONFERENCE ON DATABASE THEORY, 1997. Delphi, Greece. **Invited Paper**. [S.l.: s.n.] p. 1-18. Disponível em: <<http://www.rocq.inria.fr/~abitebou/pub/index.html>> Acesso em: 23 set. 2002.
- [BAT 86] BATINI, C.; LENZERINI, M.; NAVATHE, S.B. A Comparative Analysis of Methodologies for Database Schema Integration. **ACM Computing Surveys**, New York, p. 324 – 364, 1986.
- [BAT 92] BATINI, C.; CERI S.; NAVATHE, S.B. **Conceptual Database Design**. [S.l.]: The Benjamin Cummings, 1992.
- [BEC 93] BECKWITH, Richard; MILLER, George A. Tengi Randee. **Design and implementation of the WordNet Lexical Database and searching software**. Disponível em: <<http://www.cogsci.princeton.edu/~wn/papers.shtml>>. Acesso em: 2 mar. 2001.
- [BEN 99] BENJAMINS, V. R.; FENSEL, D.; DECKER, S.; GÓMEZ-PÉREZ, A. (KA)2: Building Ontologies for the Internet: a Mid Term Report. **International Journal of Human-Computer Studies**, [S.l.], v.51, p.687-712. 1999. Disponível em: <http://www.swi.psy.uva.nl/usr/richard>. Acesso em: 22 mai. 2000.
- [BER 99] BERGAMASCHI, S. et al. **Intelligent Techniques for the Extraction and Integration of Heterogeneous Information**. 1999. Disponível em: <<http://sunsite.informatik.rwth-aachen.de/Plublication/CEUR-WS>>. Acesso em: 24 ago. 2001.
- [BLÁ 98] BLÁZQUEZ, M. et al. **Building Ontologies at the Knowledge Level Using the Ontology Design Environment**. Madrid: Laboratorio de Inteligencia Artificial Universidad Politécnica de Madrid, 1998.
- [BON 94] BONJOUR, Michel; FALQUET, Gilles. **Concept Bases: A Support to Information Systems Integration**. CaiSe 94 Conference, Utrech. 1994. Disponível em: <http://cui.unige.ch/db-research/members/mb/papers/caise94/CAISE94_1.html>. Acesso em: 12 fev. 2002.
- [BOU 99] BOUGUETTAYA, Athman; BENATALLAH, Boualem; ELMAGARMID, Ahmed. **Management of Heterogeneous and Autonomous Database Systems**. New York: Morgan Kaufmann, 1999.
- [CAS 99] CASTILHO, Mauro J.M.V. de et al. The SIDI Health System Project. In: PROTEM-CC - INTERNATIONAL EVALUATION, 2., 1999, Rio de Janeiro. **Proceedings...** Brasília: CNPq, 1999. p.407-420.
- [COV 2002] COVER, Robin. **Managing Names and Ontologies: An XML Registry and Repository**. 1999. Disponível em: <<http://www.oasis-open.org/cover/xmlArticles>>. Acesso em: 24 fev. 2002.
- [DEC 98] DECKER, Stefan et al. **Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information**. Karlsruhe: Institute AIFB, University de Karlsruhe. [S.l.]: Kluwer Academic Publishers, 1998.
- [DES 86] DESOUSA, J. M. SIS - A Schema Integration System. In: BRITISH NATIONAL CONFERENCE ON DATABASES, 5., 1986. **Proceedings...** [S.l.: s.n.], 1986.

- [DOR 2000] DORNELES, Carina F. **Extração de Dados Semi-Estruturados com Base em uma Ontologia**. 2000. Dissertação de Mestrado em Ciência da Computação. Instituto de Informática, UFRGS, Porto Alegre. Disponível em: <<http://metropole.inf.ufrgs.br/groupPublications>>.
- [DUS 97] DUSCHKA, Oliver M.; GENESERETH, Michael R. Infomaster – an Information Integration Toll. In: INTERNATIONAL WORKSHOP ON INTELIGENT INFORMATION INTEGRATION, 1997. **Proceedings...** [S.l.: s.n.], 1999.
- [ELM 99] ELMAGARMID, Ahmed; DU, Weimin; AHMED, Rafi. **Management of Heterogeneous and Autonomous Database Systems**. New York: Morgan Kaufmann, 1999.
- [EMB 99] EMBLEY, David W. et al. Ontology suitability for Uncertain Extraction of Information from Multi-Record Web Documents. In: ADI, 1999. **Proceedings...** Disponível em: <<http://www.deg.byu.edu/papers>>. Acesso em: 18 jul. 2001.
- [FAN 91] FANG, D. et al. Remote-Exchange: An approach to controlled sharing among autonomous, heterogeneous database system. In: IEEE SPRING COMPCON, 1991. **Proceedings...** Los Alamitos, CA: [s.n.], 1991.
- [FAR 96] FARQUHAR, Adam; FIKES, Richard; RICE, James. **The Ontolingua Server: a Tool for Collaborative Ontology Construction**. LA: Knowledge System Laboratory, Stanford University, 1996.
- [FEN 99] FENSEL, Dieter et al. **ON2BROKER: Semantic-Based Access to Information Sources at the WWW**. Karlsruhe: Institute AIFB, University of Karlsruhe, 1999.
- [FER 97] FERNÁNDEZ, Mariano; GÓMEZ-PÉRES, Asunción; JURISTO, Natalia. **Methontology: from Ontological Art Towards Ontological Engineering**. Madrid: Laboratorio de Inteligencia Artificial, Universidad Politécnica de Madrid, 1997.
- [GEN 97] GENESERETH, Michael R.; KELLER, Arthur M.; DUSCHKA, Oliver M. **Infomaster: an Information Integration System**. **SIGMOD Record**, New York, v. 26, n. 2, June 1997. Trabalho apresentado na ACM SIGMOD International Conference on Management of Data, 1997.
- [GOM 96] GÓMEZ-PÉREZ, Asunción; FERNÁNDEZ, Mariano; VICENTE, Antonio J. Towards a Method to Conceptualize Domain Ontologies. Madrid: Laboratorio de Inteligencia Artificial, Universidad Politécnica de Madrid, 1996.
- [GOT 92] GOTTHARD, W.; LOCKEMANN, P.C.; NEUFELD, A. System-Guided View Integration for Object-Oriented Databases. **IEEE Transactions on Knowledge and Data Engineering**, New York, v. 4, n. 1, p. 1 - 22, 1992.
- [GRU 93] GRUBER, Thomas R. **A Translation Approach to Portable Ontology Specification**. Stanford: Knowledge Systems Laboratory, Stanford University, 1993.
- [GUA 98] GUARINO, Nicola. **Semantic Matching: Formal Ontological Distinctions Information Organization, Extraction and Integration**. Frascati, Italy: Summer School on Information Extaction, 1998.
- [HAM 99] HAMMER, Joachim; MCLEOD, Dennis. **Management of Heterogeneous and Autonomous Database Systems**. New York: Morgan Kaufmann. 1999.
- [HAY 90] HAYNE, S; RAN, S. **Multiuser View Integration System (MUVIS): An Expert System for View Integration**. In: THE INTERNATIONAL CONFERENCE ON DATA

- ENGINEERING, 6., 1990. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1990.
- [KAS 99] KASHYAP, Vipul; SHETH, Amit. **Management of Heterogeneous and Autonomous Database Systems**. New York: Morgan Kaufmann, 1999.
- [KEL 2000] KELLER, Arthur M. **Uso de Ontologias no Projeto Infomaster**. Mensagem recebida por <smsanti@terra.com.br>. em 14 mar. 2000.
- [LAR 89] LARSON, J., NAVATHE, S., ELMASRI, R. A Theory of Attribute Equivalence in Databases with Application to Schema Integration. **IEEE Transactions on Software Engineering**, New York, v. 15, n. 4, p. 449-463, 1989.
- [LIN 2001] LIM, S.; NG, Y. An Automated Integration Approach for Semi-Structured and Structured Data. In: INTERNATIONAL SYMPOSIUM ON COOPERATIVE DATABASE SYSTEM FOR ADVANCED APPLICATION, CODAS, 3., 2001, Beijing. **Proceedings...** [S.l.: IEEE], 2001.
- [MAR 2000] MARTINS, Luís A. **Uma Ferramenta para Integração de Esquemas XML utilizando Ontologias**. 2000. Trabalho de Graduação em Informática (Ciência da Computação). Universidade Federal de Pelotas, Pelotas.
- [MCB 2001] MCBRIEN, P.; POULOVASSILIS, A. A Semantic Approach to integrating XML and Structured Data Sources. In: CONFERENCE ON ADVANCED INFORMATION SYSTEM ENGINEERING, CAISE, 13., 2001. **Proceedings...** Interlaken, Switzerland: Springer Verlag, 2001. p. 330-345.
- [MEL 2000] MELLO, Ronaldo S.; DORNELES, Carina F.; KADE, Adrovane; BRAGANHOLO Vanessa P.; HEUSER, Carlos A. Dados Semi-Estruturados. In: SBBD, 2000. **Tutorial**. Disponível em: <<http://metropole.inf.ufrgs.br/groupPublications>>. Acesso em: 25 dez. 2000.
- [MEL 2001] MELLO, Ronaldo S.; HEUSER, Carlos A. Aplicação de Ontologias a Dados Semi-Estruturados. 2001. Universidade Federal do Rio Grande do Sul. Disponível em: <<http://metropole.inf.ufrgs.br/groupPublications>>. Acesso em: 26 dez. 2001.
- [MEL 2002] MELLO, Ronaldo S. Uma Abordagem Bottom-Up para Integração Semântica de Esquemas XML. 2002. Tese (Doutorado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em: <<http://metropole.inf.ufrgs.br/groupPublications>>. Acesso em: 23 ago. 2002.
- [MIL 93] MILLER, George A. et al. **Introduction to WordNet: An On-Line Lexical Database**. 1993. Disponível em: <<http://www.cogsci.princeton.edu/~wn/papers.html>>. Acesso em: 2 abr. 2002.
- [PIM 2000] PIMENTEL, M.; TEIXEIRA, C.; SANTANCHÈ A. **XML: Explorando suas aplicações na Web**. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA, 2000, Curitiba. **Anais...** Curitiba: Ed. Champagnat, 2000. p. 1-43.
- [RAM 95] RAMASH, V.; RAN, S. A Methodology for Interschema Relationship Identification in Heterogeneous Databases. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEMS AND SCIENCES, 3., 1995. **Proceedings...** [S.l.: s.n.], 1995.
- [RAM 97] RAMASH, V.; RAN, S. Integrity Constraint Integration in Heterogeneous Databases: An Enhanced Methodology for Schema Integration. **Information Systems**, Oxford, v. 22, n. 8, p. 423-446, 1997.

- [RAS 99] RAM, Sudha; RAMASH, V. **Management of Heterogeneous and Autonomous Database Systems**. New York: Morgan Kaufmann, 1999.
- [RAN 95] RAN, S.; RAMASH, V. A Black Board Based Cooperative System for Schema Integration. **IEEE Expert**, Los Alamitos, v. 10, n. 3, p. 56-63, 1995.
- [REY 2001] REYNAUD, C.; SIROT, J.; VODISLAV, D. Semantic Integration of XML Heterogeneous Data Sources. In: INTERNATIONAL DATABASE ENGINEERING & APPLICATIONS SYMPOSIUM, IDEAS, 2001, Grenoble France. **Proceedings...** [S.l.: IEEE], 2001.
- [ROD 2001] RODRIGUEZ-GIANOLLI, P.; MYLOPOULOS, J. A Semantic Approach to XML-Based Data Integration. In: INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING, ER, 20., 2001, Yokohama, Japan. **Proceedings...** [S.l]: Springer-Verlag, 2001. p 117-132.
- [SAN 2000] SANTI, Sergio M. **Ontologias - Abordagens de Construção e Aplicações**. Trabalho Individual (Mestrado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em: <<http://metropole.inf.ufrgs.br/groupPublications>>. Acesso em 28 fev. 2002.
- [SHE 88] SHETH A. et al. A Tool for Integrating Conceptual Schemata and User Views. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING. 1988. **Proceedings...** [S.l.: IEEE Computer Society Press], 1988.
- [SHE 90] SHETH, A.; LARSON, J. Federated database systems for managing distributed, heterogeneous, and autonomous databases. **ACM Computing Surveys**, New York, v. 22, n. 5, p. 183-236, 1990.
- [SHO 91] SHOVAL, P.; ZOHN, S. Binary Relationship Integration Methodology. **Data and Knowledge Engineering**, New York, v. 11, n. 6, p. 225-250, 1991.
- [SIL 2002] SILBERSCHATZ, Abraham.; KORTH, Henry F.; SUDARSHAM S. **Database Systems Concepts**. 4th. ed. New York: McGraw-Hill, 2002. p. 225-250.
- [STU 98] STUDER, Rudi; BENJAMINS, V. Richard; FENSEL, Dieter. Knowledge Engineering: Principles and Methods. **Data & Knowledge Engineering**, Karlsruhe, v.25, n. 5, p. 161-167, 1998. Disponível em: <<http://www.aifb.uni-karlsruhe.de/WBS/publications/index.html>>. Acesso em: 22 dez. 2000.
- [STU 99] STUDER, Rudi et al. Knowledge Engineering: Survey and Future Directions. 1999. Disponível em: <<http://www.aifb.uni-karlsruhe.de/WBS/publications/index.html>>. Acesso em: 22 dez. 2000.
- [STU 2000] STUDER, Rudi; ERDMANN, M. **How to Structure and Access XML Documents with Ontologies**. Karlsruhe: Institut Für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) – Universidade de Karlsruhe, 2000. Disponível em: <<http://www.aifb.uni-karlsruhe.de/WBS/publications/index.html>>. Acesso em: 22 dez. 2000.
- [USC 96] USCHOLD, M.; GRUNINGER, M. Ontologies: Principles, Methods and Applications. **KNOWLEDGE ENGINEERING REVIEW**, Stanford, v. 11, n. 2, 1996.
- [VDO 2001] VDOVJAK, R.; HOUBEN, G. RDF-Based Architecture for Semantic Integration of Heterogeneous Information Sources. In: INTERNATIONAL WORKSHOP ON INFORMATION INTEGRATION ON THE WEB, WIIW, 2001, Rio de Janeiro, Brasil. **Anais...** [S.l.]: UNIRIO, 2001. p 51-57.
- [W3C 2002] XML. Disponível em: <<http://www.w3.org/XML>>. Acesso em: 31 jun. 2002.